AD 744 082

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R & D		
(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)		
1. ORIGINATING ACTIVITY (Corporate author) Philco-Ford Corporation Communications & Technical Services Division Willow Grove, PA 19090	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP N/A	
3. REPORT TITLE ALGORITHMS FOR TRANSCEIVER MODULATION AND DEMODULATION		
4. DESCRIPTIVE NOTES (Type of report and inclusive dates) Final Report February 1970 to August 1971		
5. AUTHOR(S) (First name, middle initial, last name) Kenneth Abend, David L. Fletcher, Constantine Gamacos, W. Andrew Wright		
6. REPORT DATE January 1972	7a. TOTAL NO. OF PAGES 230	7b. NO. OF REFS 27
8a. CONTRACT OR GRANT NO. F30602-70-C-0052 Job Order No. 45190000	9a. ORIGINATOR'S REPORT NUMBER(S) Philco-Ford Number 2580 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) RADC-TR-71-305	
10. DISTRIBUTION STATEMENT Approved for public release; distribution unlimited.		
11. SUPPLEMENTARY NOTES None	12. SPONSORING MILITARY ACTIVITY Rome Air Development Center (CORS) Griffiss Air Force Base, New York 13440	

13. ABSTRACT

This report is devoted to the investigation of algorithms which reduce the complexity of a real-time digital processor for performing tactical transceiver functions of modulation, demodulation, frequency synthesis, heterodyning, and filtering. A minimum complexity design for a Multimode Digital Processing Transceiver was established through analytical investigations and computer simulations, and a breadboard was developed for experimental evaluation.

DD FORM 1 NOV 65 1473

UNCLASSIFIED
Security Classification

UNCLASSIFIED
Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Digital Transceivers						
Communication Systems						
Bandpass Sampling						
Modulation and Demodulation						

UNCLASSIFIED
Security Classification

ALGORITHMS FOR TRANSCEIVER MODULATION AND DEMODULATION

Kenneth Abend
David L. Fletcher
Constantine Gamacos
W. Andrew Wright

Philco-Ford Corporation

ib

FOREWORD

This Final Report was prepared by Kenneth Abend, Constantine Gumacos, and W. Andrew Wright of the Advance Technology Department, Communications and Technical Services Division, Philco-Ford Corporation, Willow Grove, Pennsylvania. The computer simulations were performed and documented by David L. Fletcher. The breadboard (hardware and software) was designed, programmed, and tested by Mr. Wright.

The report was submitted under Contract F30602-70-C-0052, Job Order No. 45190000, for Rome Air Development Center, Griffiss Air Force Base, New York. Philco-Ford number is 2580. Period covered is 20 February 1970 to 6 August 1971. Edward E. Cossette (CORS) was the RADC Project Engineer in charge.

This report has been reviewed by the Information Office (OI) and is releasable to the National Technical Information Service (NTIS).

This technical report has been reviewed and is approved.

Approved: Edward E Cossette
EDWARD E. COSSETTE
Effort Engineer

Approved: Al Twitchell
ALVIN TWITCHELL, Colonel, USAF
Chief, Communications & Navigation Division

FOR THE COMMANDER: Fred I Diamond
FRED I DIAMOND
Acting Chief, Plans Office

ABSTRACT

This report is devoted to the investigation of algorithms which reduce the complexity of a real-time digital processor for performing tactical transceiver functions of modulation, demodulation, frequency synthesis, heterodyning, and filtering. A minimum complexity design for a Multimode Digital Processing Transceiver was established through analytical investigations and computer simulations, and a breadboard was developed for experimental evaluation.

EVALUATION

This study was concerned with further development of the multimode digital transceiver concept which was specified in a previous study. This concept allows replacement of major portions of the analog circuits in a transceiver with logic circuitry in the form of a special purpose digital processor. This concept would allow acceptance of analog data such as voice or digital data to be converted through appropriate sampling and quantizing to a numerical representation of these inputs for subsequent modulation in nearly any known form (i.e. analog AM, AM-SSB, FM, etc. or digital data PSK, FSK or combinations, etc.). Subsequently the outputs would be processed through digital translation, digital to analog conversion, further analog frequency translation as necessary, final filtering and amplification prior to transmitting. At the receiver the same digital processor reconfigured would perform the receiving functions after initial amplification, heterodyning and IF filtering. Conceptually all transmitter functions in the HF and VHF frequency bands could be accomplished numerically with the exception of final filtering and amplification. However this is not considered to be a cost effective approach at this time since the digital to analog converter would be expensive and further the digital processor could not be fully utilized for performing the receiver functions. In fact the contractor demonstrated ingenuity in developing a minimum complexity digital processor design which makes use of the same common elements for transmitting functions such as modulation, frequency synthesis, and frequency translation as for receiving functions such as demodulation and filtering. Another important aspect developed was, use in time sharing arithmetic functions in the filter design, which makes up a major part of the digital processor, and as a result reduces the complexity and in turn the cost of the digital processor substantially.

Based on the design work developed in this study and the breadboard demonstration of transmitting and receiving many modulation types with a single digital processor, it appears that the next logical step is to develop an experimental model for evaluation with many existing Air Force HF/VHF transceivers in field use today as well as demonstrating additional digital data transmission capabilities peculiar to this concept. This new signal processing concept offers the potential for an equipment transition period making use of this equipment as well as existing field equipments simultaneously. Since this single unified processor approach offers the capability of interfacing with many existing field radios, because it can be programmed to handle characteristics of these existing transceivers, a transition period for use of both the existing equipment and the new development is possible. This circumvents the problem of total replacement with new equipment but still offers additional operational characteristics not presently incorporated in the present transceivers in use today.

Edward E. Cossette

EDWARD E. COSSETTE
Effort Engineer

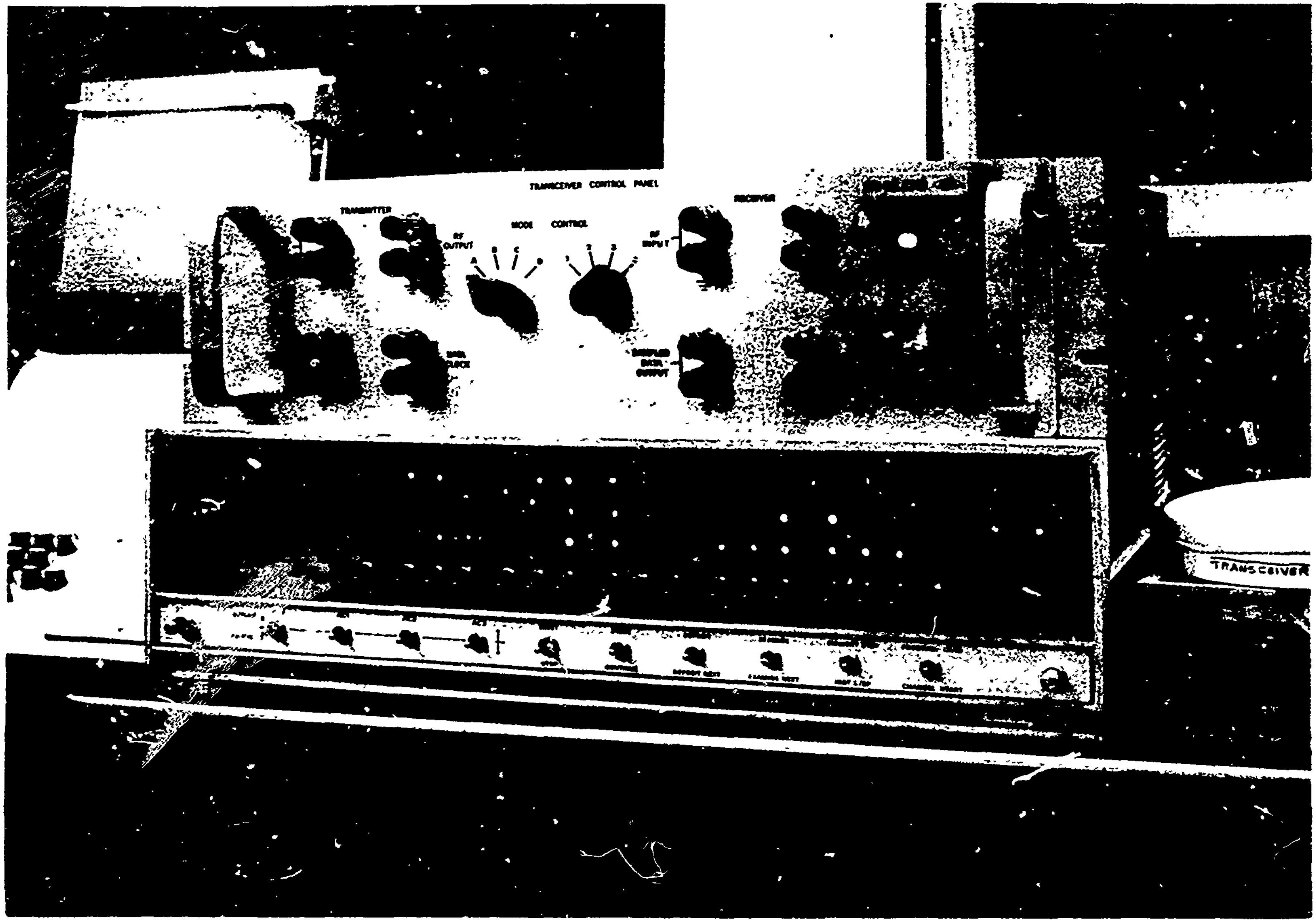
TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Continued)

LIST OF ILLUSTRATIONS (Continued)



Frontpiece: Multimode Digital Processing Transceiver--Breadboard Model

SECTION I

INTRODUCTION

This report is the latest in a series of investigations related to the use of a real-time digital processor to perform tactical transceiver functions of modulation, demodulation, frequency synthesis, heterodyning and filtering. The objective of the present program was to investigate algorithms which reduce the complexity of the processor. A minimum-complexity design for a Multimode Digital Processing Transceiver was established and a breadboard was developed for experimental evaluation.

In the Digital Equivalent Transceivers Study (Reference 1) the idea of using a numerical processor to perform transceiver functions was introduced and alternative approaches to performing these functions were investigated. In the present effort, these investigations were continued, design decisions were made, and specific computational algorithms were developed and tested. Use was made of results from some of our previous investigations[1, 20, 27], while duplication of material therein, was avoided. The transceiver specifications in Section V.2 of Reference 1 were used as guides and goals through the present work.

1. DIGITAL TRANSCEIVERS

A Digital Processing Transceiver interfaces with the analog world through analog-to-digital (A/D) and digital-to-analog (D/A) converters at voiceband and at an intermediate frequency (IF). Someday, perhaps, the latter interface may be at RF. Between the interfaces, all filtering, spectral shaping modulation, demodulation, heterodyning coding, and decoding for voice and for digital data is performed by time sharing a special purpose, high speed, digital processor.

The signals to be processed are quantized in time and amplitude and the processor operates as a real-time digital computer on a sequence of numbers. By switching between several "hardwired" programs the processor becomes: either a receiver or a transmitter; for either voice or digital data; via amplitude, phase, or frequency modulation; either single- or double-sideband; either coherent, non-coherent, or differentially coherent; with or without partial-response spectral shaping, encryption, scrambling, or any other type of signal processing we may wish to program.

2. SUMMARY

The overall system design that evolved from this work is presented

in Sections II.1 through II.3. The basic system involves filtering, resampling, angle modulation, and frequency translation, for transmitter operation; and bandpass sampling, frequency or phase demodulation, resampling, and filtering, for receiver operation. The interpolation resampling process (for transmitter operation) is discussed in detail in Section II.4. The discussion builds on, and extends the corresponding discussion in Section III.1 of Reference 1.

Zero differential delay (absolutely linear phase) is essential to avoid pulse distortion and resulting intersymbol interference for digital data transmission[18]. While this ideal is impossible to attain with analog filters, it is easily attained with finite response digital filters. All known methods for designing such filters (including some that have not yet been published) are discussed in Section III.

The problem of accurately extracting the complex modulated low-pass signal by sampling the received I.F. signal at a rate determined by the bandpass sampling theorem is treated in Section IV. Complex sampling by two A/D convertors, operating a quarter of a carrier-cycle apart in time, is only an approximation to true complex sampling. A method for improving the approximation, without using more than two A/D convertors or a higher sampling rate, is presented and compared with the multiple sample method of Reference 19.

Modulation and demodulation is discussed in Section V. Linear (amplitude) modulation and demodulation, whether double-sideband or single-sideband, is seen to be simply a problem in filtering and frequency translation. Frequency modulation is obtained by merely inserting the modulator of figure 20a before the frequency translator in figure 19a. Two alternative schemes for frequency (or phase) demodulation are shown in figures 21 and 11.

Recursive and nonrecursive filters are compared in Section VI and specific filter designs are given. The unit response of a half-sample delayed digital differentiator[25] is analytically derived in Section VI.3.

The entire transceiver system was simulated in the SSB, DSB, and FM mode. Spectral examinations were made via the FFT at various points in the system. Noise was introduced and signal-to-noise ratios were measured. The simulations presented in Section VII resulted in some design changes that were incorporated into the breadboard and into the system description of Section II.

The entire system, appropriately revised, was then breadboarded as a 100:1 scaled down (in frequency) version. The scaling was done so that an off-the-shelf general purpose processor could be utilized. The bread-

board included all necessary A/D and D/A converters and performed all necessary digital frequency translations. It was tested in the DSB, SSB, PM, and FM modes with both analog and digital data. Error rate measurements were made in FSK operation.

Conclusions and Recommendations are in Section IX.

SECTION II

SYSTEM DESCRIPTION

This section presents the system configuration of the multimode transceiver processor in its simplest form. The double sideband, single sideband, and angle modulation system are discussed separately for the sake of clarity. In reading the double sideband AM discussion, which is presented first because it is the simplest, the reader may feel that the system is unduely complicated for the simple task being performed. He must bear in mind that the configuration was chosen so that the same system can be utilized for all forms of modulation.

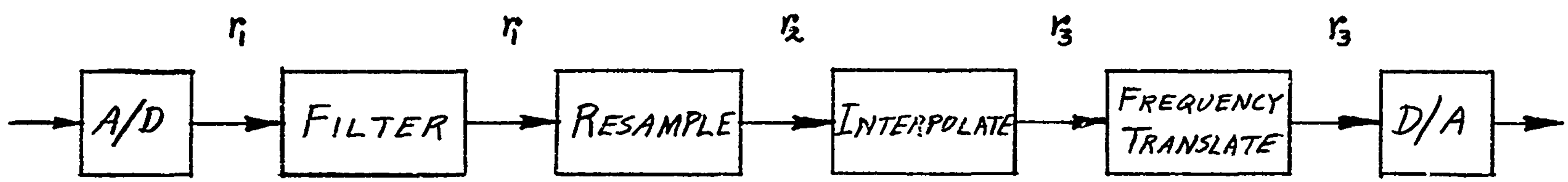
1. DOUBLE SIDEBAND AM

The double sideband amplitude modulated (DSB-AM) transceiver configuration is given in figure 1. The filters and resamplers are convolutional filters as shown in figure 2. In practice, a single shift register and correlator would be time shared for all transceiver filtering and resampling functions through appropriate timing and control circuitry. For ease of exposition, however, we will treat them as if they were separate filters.

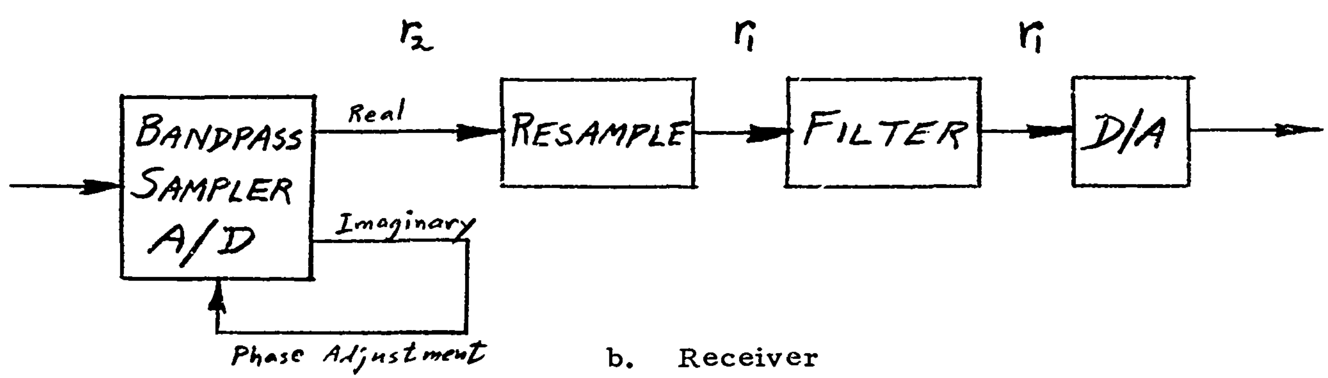
The first unit in figure 1a is an analog to digital-sampled-data (A/D) converter. The input is a baseband analog signal and the output is a sequence of binary numbers representing the sampled values of the input taken at the rate of r samples per second. Figure 3a shows a representative baseband signal spectrum where, because of RF spectral crowding, we desire to transmit only B hertz of the signal. Typically, B will be in the order of 3 kilohertz. Figure 3b shows the spectrum at the output of the A/D converter. Since the spectrum repeats at the sampling rate, * that rate must be chosen greater than twice the total signal bandwidth ($r > 2B'$) in order to prevent aliasing. However; aliasing is permissable as long as it does not encroach on the desired portion of the spectrum (figure 6b). Thus the sampling rate need only satisfy $r \geq B + B'$.

The spectra depicted in figure 3 assumes a sampling rate of $r_1 = r_2 = r = 16$ kilohertz, corresponding to figure 1a with the resampling filter (the third unit in the figure) removed. The filter (the second unit) is non-

*Since periodic time signals give rise to discrete spectral components, by the symmetry of time-frequency relations, discrete time signals give rise to periodic spectra.

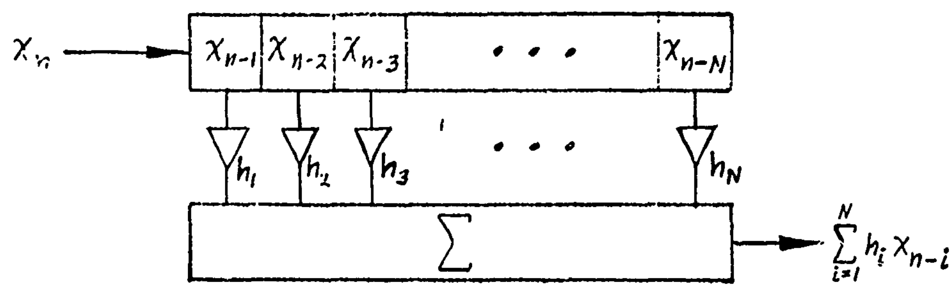


a. Transmitter

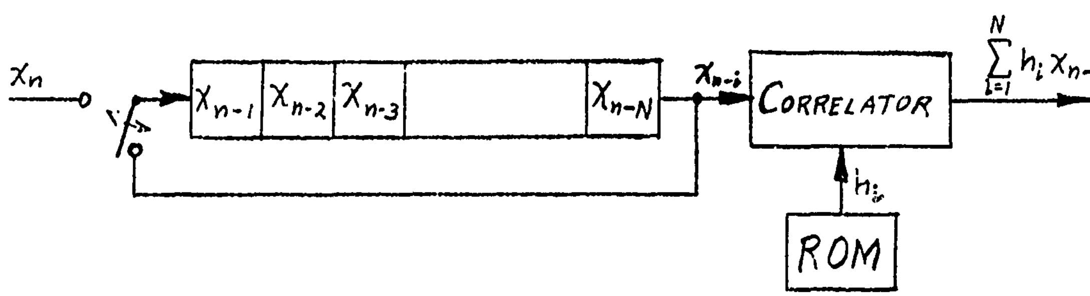


b. Receiver

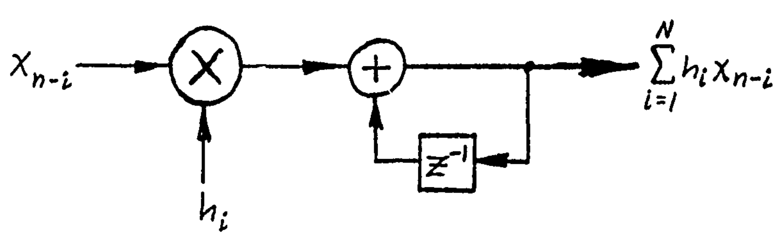
Figure 1: DSB-AM Transceiver Configuration



a. Schematic of Convolutional Filter



b. Hardware for Convolutional Filter



c. Schematic of Correlator

Figure 2: The Convolutional Filter

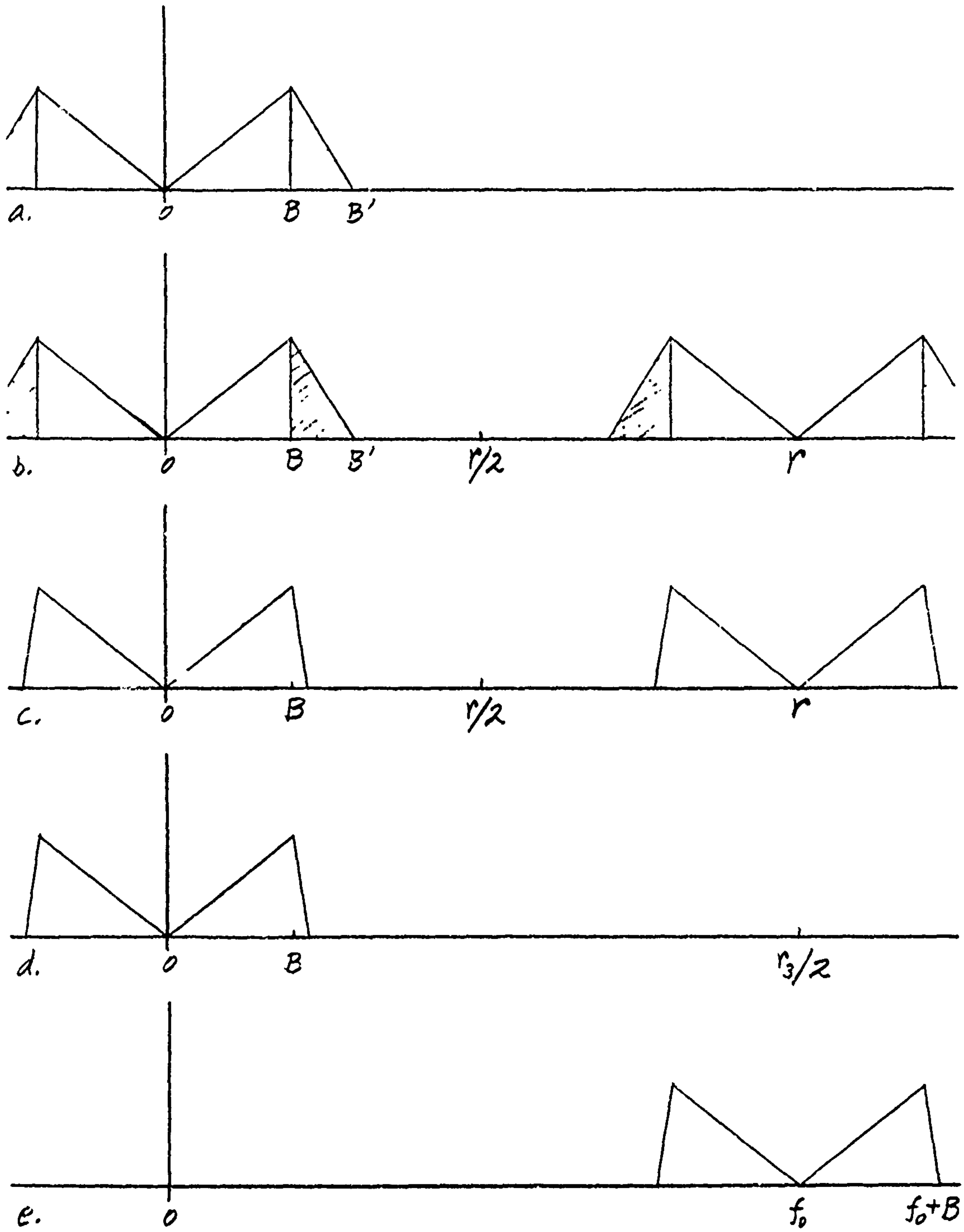


Figure 3: DSB-AM Transmitter Spectra
(without resampling filter; Symmetric about zero frequency)

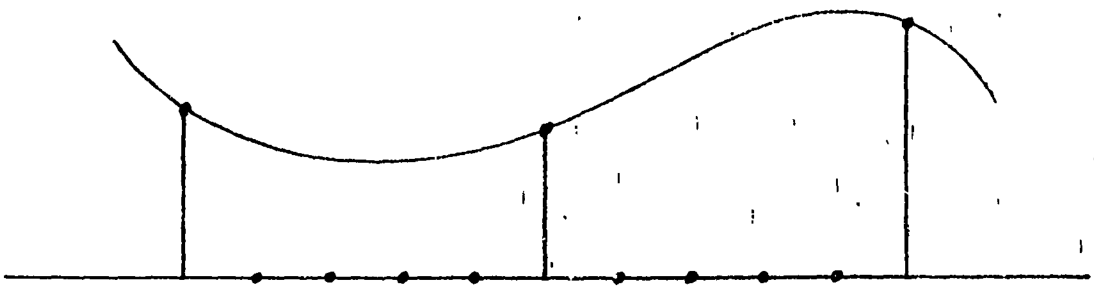
recursive (convolutional) because of the straight forward implementation and because it can have zero differential delay (perfectly linear phase) resulting in undistorted pulses when the system is used for digital data communications, instead of for voice.

The convolutional filter hardware (figure 2b) operates at N times the sampling rate, where N is the number of stages in the filter. The N filter tap weights are stored in an MOS read-only-memory (ROM). After every N shifts of the recirculating register, a new sample is shifted in, the oldest sample is shifted out, a result is read out of the correlator, and the accumulator in the correlator is reset.

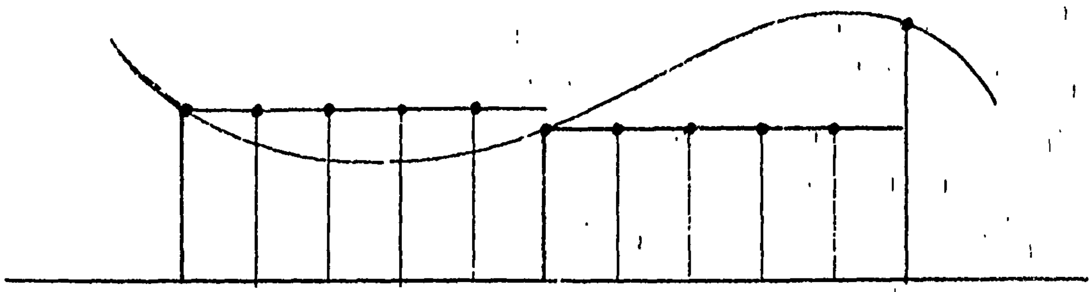
In this simple version (no resampling filter) the output of the filter is the input to the interpolator (figure 1a) and has a spectrum as shown in figure 3c. The interpolator serves to raise the sampling rate to twice the I. F. carrier frequency ($r_3 = 2f_o$). Linear interpolation is utilized because it combines implementation simplicity with adequate suppression of unwanted spectral repeats. Filling in with zero samples as in figure 4a would not change the spectrum from that of figure 3c. The store and repeat interpolation of figure 4b would attenuate the first spectral repeat by the first zero crossing of a (sin x)/x curve. While linear interpolation (figure 4c) attenuates it more effectively; it is still imperative that the input sampling rate, r_2, be high enough for the interpolator to adequately attenuate the first spectral repeat.

After interpolation (figure 3d), the samples are multiplied by $\cos(2\pi f_o n/r_3) = \cos(\pi n) = (-1)^n$ resulting in the spectrum of figure 3e. This negation of every other sample is the digital equivalent of frequency translation to the intermediate carrier frequency, $f_o = 2r_3$. Digital frequency synthesis and translation to RF are discussed elsewhere [1, 2].

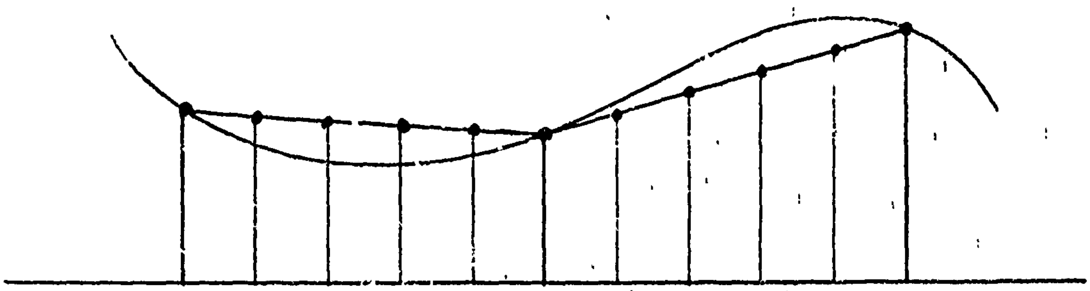
The corresponding DSB-AM receiver is as shown in figure 1b, but again with the resampling filter removed. The bandpass sampler operates on a relatively wide-band IF signal as represented by figure 5a. Note that if the signal whose spectrum is illustrated in 5a is sampled at the carrier frequency ($r = f_o$) or half the carrier frequency ($r = f_o/2$), or any integer fraction of the carrier frequency ($r = f_o/k$), the resulting spectrum has a component at zero frequency as illustrated in figure 5b. The sampling rate must satisfy the inequality $r \geq B + B''$ to prevent distortion of the desired portion of the spectrum by aliasing. Phasing of the sampler can be aided (especially in the suppressed carrier case) by an error control loop that drives the quadrature samples to zero. The quadrature (or imaginary) samples are taken one quarter of a carrier cycle after the corresponding in phase (or real) samples. The convolutional filter produces a signal with the spectrum of figure 5c which goes into a digital to analog converter producing an analog signal with spectrum given by figure 5d. Note that the filter in



a. Zero Insertion



b. Store and Repeat



c. Linear Interpolation

Figure 4: Simple Interpolation Schemes for Increasing the Sampling Rate

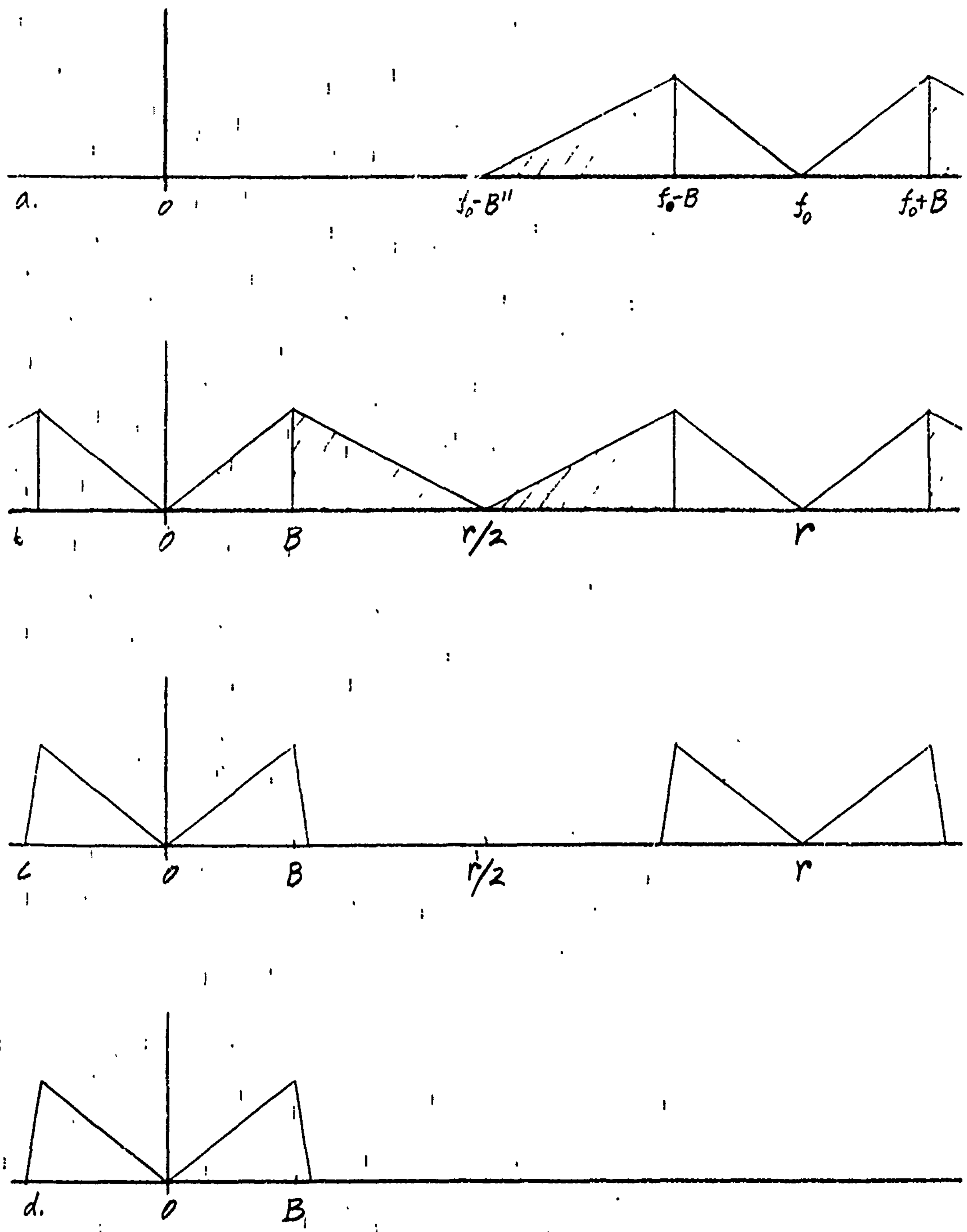


Figure 5: DSB-AM Receiver Spectra
(without resampling filter; Symmetric about zero frequency)

the receiver is identical to the filter in the transmitter, with even the same tap weights.

Figure 6 applies to figure 1a with the resampling filter in place. The initial sampling rate $r_1 = 8$ kilohertz, allows for some aliasing (figure 6b), but not enough to interfere with the desired signal ($r_1 \geq B+B'$). The resampling filter is a convolutional filter operating at $r_2 = Kr_1$ kilohertz on a filtered signal whose spectrum is shown in figure 6c. The resampling filter is designed to operate at a rate r_2, to attentuate the r_1 spectral repeats (figure 6d). However, since the input can be thought of as being zero filled, as in figure 4a, the actual resample multiplication rate is reduced by a factor of K.

By using a reduced sampling rate, $r_1 = r_2/K$ for the initial filter, the number of tap weights, as well as the processing rate of the filter is reduced by a factor of K; thus reducing the multiplication rate required for the filter by a factor of K^2. The resampling filter does not have cut-off requirements as sharp as the previous filter; it is necessary only because linear interpolation would be inadequate to properly attenuate the r_1 spectral repeats. The r_2 spectral repeats are attenuated by the linear interpolator (figure 4c) which can be thought of as a simplified second resampling filter operating at a high output sampling rate, $r_3 = 2f_o$ (figure 6e). Hardware considerations might make it more economical to increase r_2 (and hence the complexity of the resampling filter) so that the interpolator can be reduced to a simple sample and hold arrangement (figure 4b). Translation to f_o (figure 6f) is again performed by negating every other sample.

An alternative frequency translation scheme would be to convert to analog at r_2 (assuming f_o is an integer multiple of r_2) and use an analog bandpass filter to obtain the spectrum of figure 6f directly from the spectrum of figure 6d. This alternative procedure is not used because our purpose is to replace analog processing with digital processing wherever possible.

Referring to figure 1b; figures 7b, 7c, 7d, and 7e represent the spectra at the output of the sampler, the resampling filter, the filter, and the digital to analog converter, respectively. The resampling filter, as well as the regular filter, is identical in both transmitter and receiver. In receiver operation, if $r_2 = Kr_1$, the signal is loaded into the resampling filter K samples at a time so that the actual multiplication rate is again determined by the lower sampling rate.

2. SINGLE SIDEBAND AM

The single sideband amplitude modulated (SSB-AM) transceiver configuration is given in figure 8. The spectra at the input and output of the

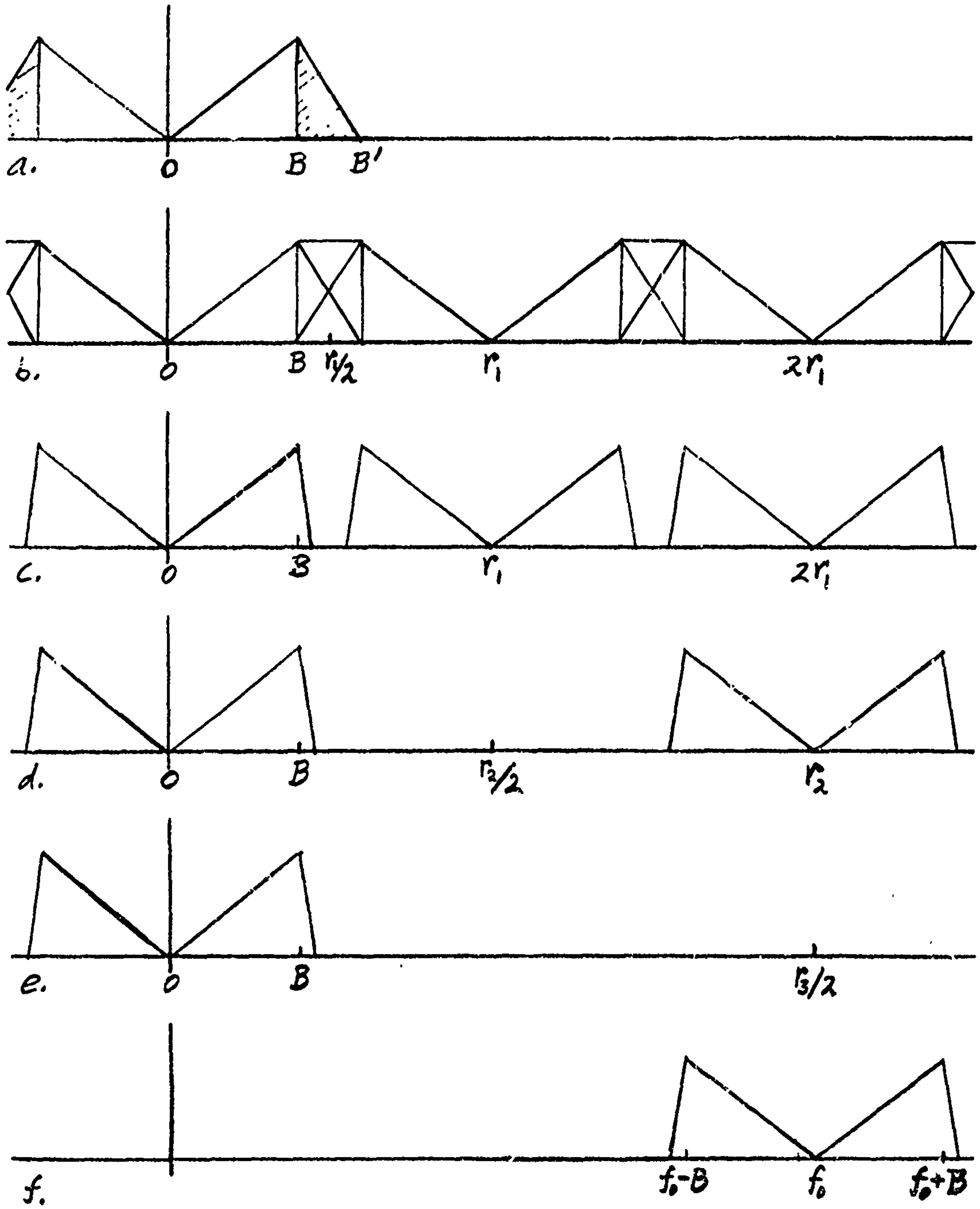


Figure 6: DSB-AM Transmitter Spectra
(Symmetric about zero frequency)

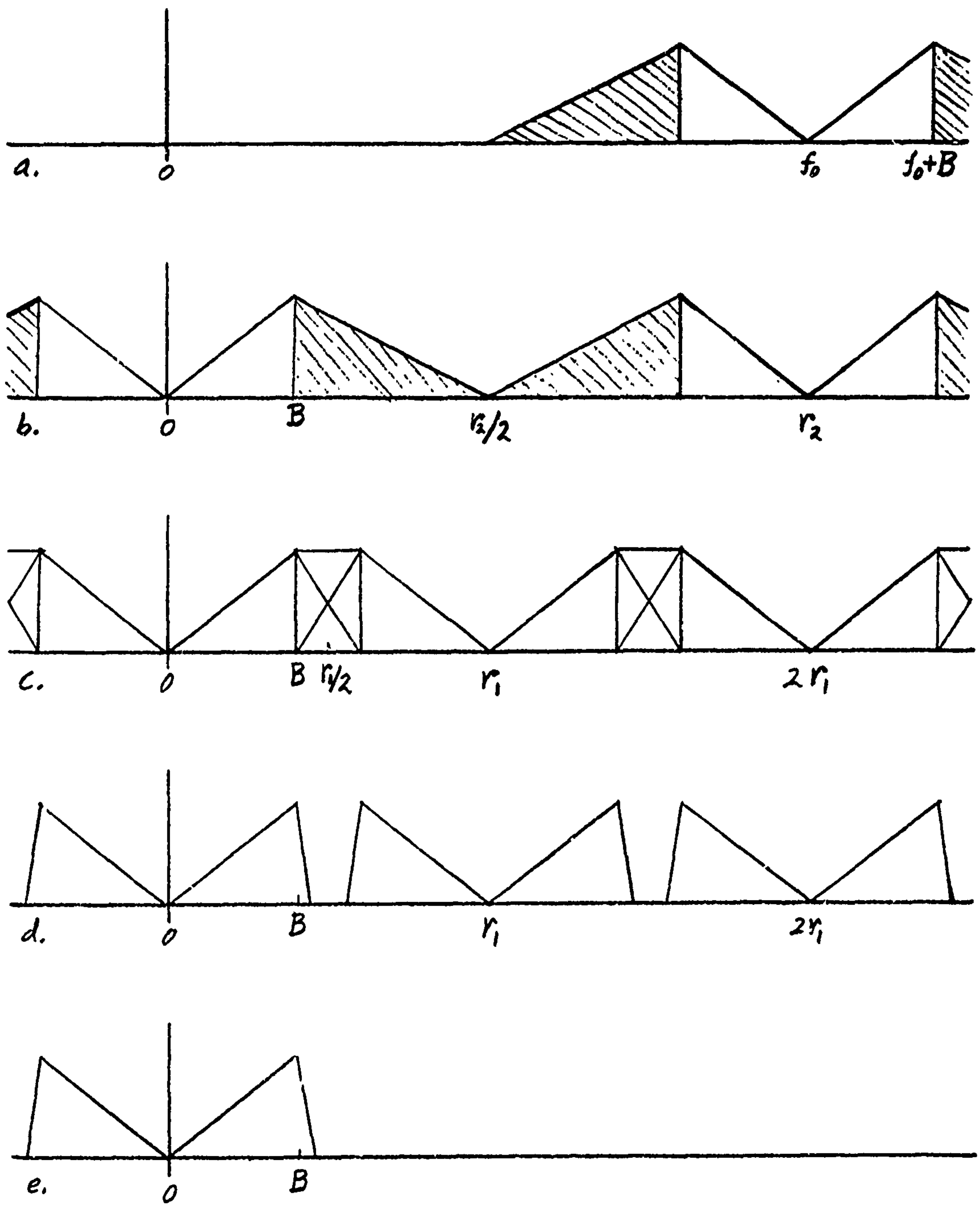


Figure 7: DSB-AM Receiver Spectra
(Symmetric about zero frequency)

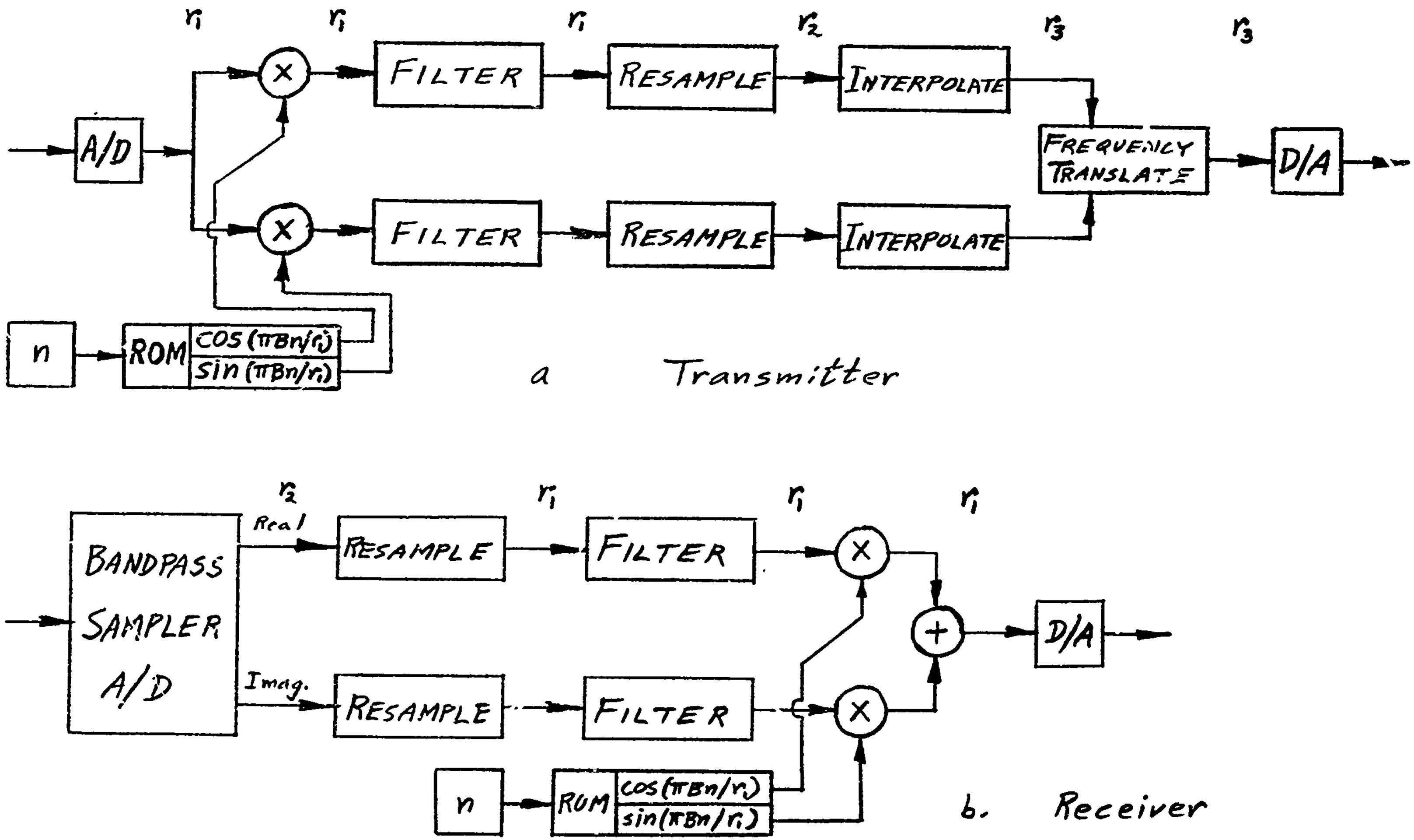


Figure 8: SSB-AM Transceiver Configuration

initial analog to digital converter are illustrated in figures 9a and 9b, assuming a sampling rate of about 8 kilohertz. As shown in figures 9c and 9d, the unwanted sideband can be filtered out by a real lowpass filter provided the spectrum is first appropriately shifted in frequency. A shift to the left (or right) by B/2 hertz, for upper (or lower) sideband modulation, is accomplished by multiplying the n^{th} sample by $e^{\mp j\pi Bn/r_1} = \cos(\pi Bn/r_1) \mp j \sin(\pi Bn/r_1)$. The resulting complex signal (figure 9c) is then filtered (figure 9d), resampled, and interpolated to produce the spectrum of figure 9e at a sampling rate of $r_3 = 4f_o$, where $f_o = f_c \pm B/2$, and f_c is the nominal IF carrier frequency. The spectra in figures 9c, 9d, and 9e are not symmetric about zero frequency and hence the corresponding time signals are complex. The upper branch in figure 8a corresponds to the real part of the signal and the lower branch, the imaginary part.

The frequency translator operates at $r_3 = 4f_o$ samples per second on a complex input signal

$$Z_n = X_n + jY_n \tag{1}$$

to produce the output

$$W_n = \mathrm{Re}\left[Z_n e^{j2\pi f_o n/r_3}\right] = \mathrm{Re}\left[Z_n e^{j\pi n/2}\right] = \mathrm{Re}\left[j^n Z_n\right] \tag{2}$$

whose spectrum is illustrated in figure 9f. Since W_n is real, its amplitude spectrum is symmetric about zero frequency.

Since

$$\int X(\tau) e^{j\pi B\tau} h(t-\tau) d\tau = \int X(t-\tau) e^{-j\pi B\tau} h(\tau) d\tau e^{j\pi Bt}, \tag{3}$$

an alternative to figure 8a would be to omit the read-only-memory sine and cosine generator, and instead, use different tap weights on the two filters. If h_n represents the n^{th} filter tap weight in figure 8a, the alternate system would not have the ROM or the multipliers, but instead, will have filters in the real and imaginary channels whose n^{th} tap is $h_n \cdot \cos(\pi Bn/r_1)$ and $\pm h_n \cdot \sin(\pi Bn/r_1)$, respectively. The corresponding set of spectra would be similar to figure 9, but without the shift to the left by B/2 hertz. The IF carrier in this case would be equal to $f_o = r_3/4$.

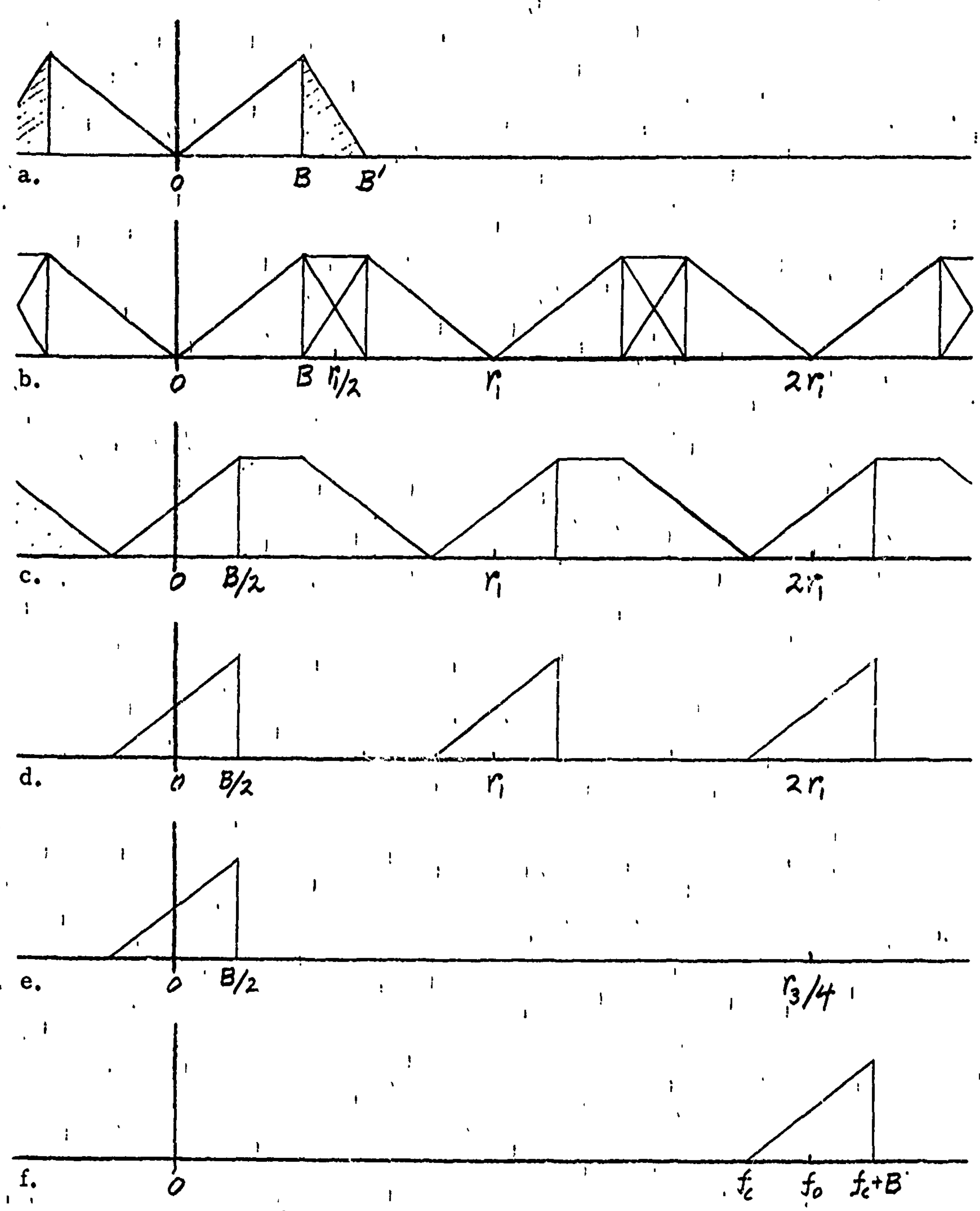


Figure 9: SSB-AM Transmitter Spectra

An advantage of this alternative scheme is that the tap weights for the real channel filter can also be used in the double sideband configuration. Since the single-sideband filter passband will be from 300 to 3,000 hertz (instead of 0 to 3,000 as illustrated), the resulting double-sideband filter will not pass d.c. This is desirable for filtering out the carrier (when operating in a non suppressed carrier mode), and is in fact, precisely how the actual double sideband filter is designed. Disadvantages of the alternative scheme are the two sets of filter tap weights are needed and, if the resampling filter remains real, its complexity is increased due to the doubling of the desired passband.

The single side sideband receiver configuration of figure 8b employs complex sampling of the real IF signal whose spectrum (symmetrical about zero frequency) is shown in figure 10a. The sampling rate, r_2, must be an integer fraction of f_o, $f_o = Kr_2$, and must be high enough to prevent aliasing from interfering with the desired signal. The digital complex signal at the output of the bandpass sampler has a spectrum such as shown in figure 10b, with spectral repeats at the sampling rate, but without symmetry. The imaginary samples may be obtained by sampling $1/(4f_o)$ seconds after the corresponding real sample, provided $f_o >>> B$. If the IF carrier is not sufficiently high, more sophisticated techniques must be employed to obtain the proper complex samples. The resampling filter brings the sampling rate down to r_1 = 8 kilohertz while cutting out the portion of the unwanted spectrum that might cause harmful aliasing at this lower rate (figure 10c). The sharp-cutoff filter can then operate at this lower sampling rate producing the output spectrum of figure 10d. The read-only-memory, multipliers and adder in figure 8b perform the functions of (1) multiplying by $e^{\pm j\pi Bn/r_1}$ to shift the spectrum B/2 hertz to the right (or left) and (2) taking the real part to make the resulting spectrum symmetrical about zero frequency (figure 10e):

$$W_n = \left[\mathrm{Re}\; Z_n e^{\pm j\pi Bn/r_1}\right] = X_n \cos(\pi Bn/r_1) \mp Y_n \sin(\pi Bn/r_1). \qquad (4)$$

After digital-to-analog conversion, the spectrum appears as in figure 10f.

3. ANGLE MODULATION (PHASE AND FREQUENCY)

The first four units of the phase or frequency modulated transmitter in figure 11a are identical to the corresponding units in the double-sideband AM transmitter. Consequently, the spectra at the five points from the input to the A/D convertor to the output of the interpolator are the same as in figures 6a through 6e, but with $r_3 = 4f_o$. After phase modulation,

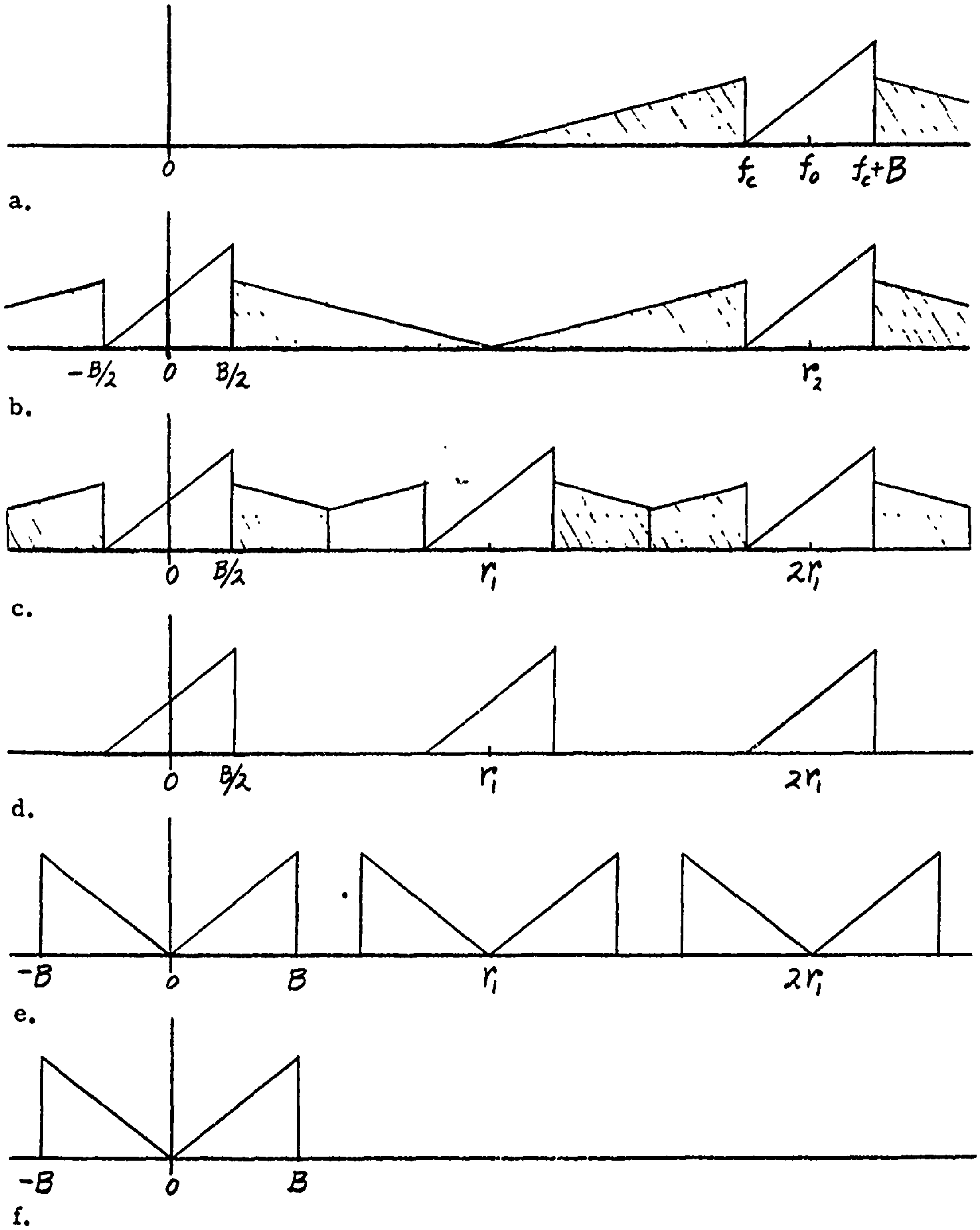


Figure 10: SSB-AM Receiver Spectra

$$Z_n = e^{jX_n} = \cos X_n + j\sin X_n, \tag{5}$$

the spectrum in figure 6e broadens and becomes nonsymmetrical. Frequency translation of this complex signal is accomplished by

$$W_n = \mathrm{Re}\left[j^n Z_n\right] \tag{6}$$

as in the case of single-sideband.

The receiver in figure 11b uses a read-only-memory at the rate r_2 to perform arc-tangent demodulation of the digital low-pass complex output of the bandpass sampler. The demodulated signal is resampled in order to reduce the complexity oi the base band filter.

Preemphasis and deemphasis are easily added to the FM configuration as illustrated in Section V.4.

4. INTERPOLATION AND RESAMPLING

All of Section III and part of Section VI will be devoted to the design of convolutional filters. In this subsection we shall consider the design and performance of the interpolation filter.

The problem, as illustrated in figure 4, is to take a signal with sampling rate r and produce a signal with sampling rate Kr (K = 5 in figure 4). Let $T = 1/(Kr)$ respresent the time between samples after interpolation. This output sampling rate, Kr, will be the basic clock rate for the interpolator.

The unit pulse is defined as

$$\delta_n = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases} \tag{7}$$

and the unit step is

$$U_n = \begin{cases} 0, & n < 0 \\ 1, & n \geq 0. \end{cases} \tag{8}$$

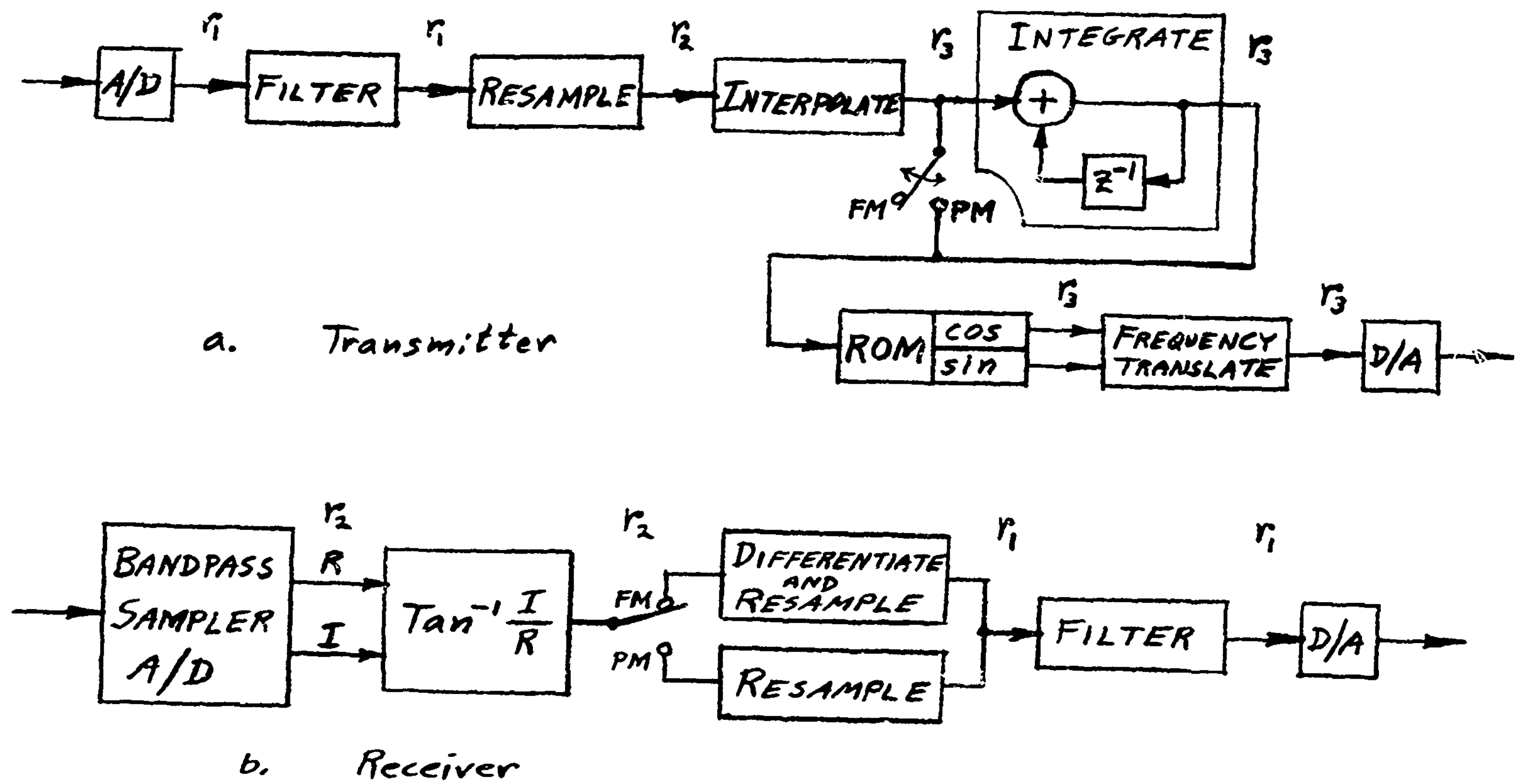


Figure 11: PM/FM Transceiver Configuration

Any function sampled at T-second intervals can be described as a superposition of unit pulses

$$x_n = \sum_i x_i \, \delta_{n-i} \tag{9}$$

The unit response of a time-invarient, linear, discrete-time system, H, is defined as its response to a unit pulse

$$h_n = H\left[\delta_n\right] \tag{10}$$

so that the output, y_n, due to an imput x_n, is given by a digital convolution:

$$y_n = H\left[x_n\right] = \sum_i x_i h_{n-i} \tag{11}$$

For a store and repeat interpolation filter (figure 4b) we have

$$y_n = \sum_{i=n-K+1}^{n} x_i \tag{12}$$

so that its unit response is given by

$$h_n = U_n - U_{n-K} \tag{13}$$

as illustrated in figure 12a. Taking the Z transform, its transfer function is

$$H(z) = \frac{1-z^{-K}}{1-z^{-1}}. \tag{14}$$

This interpolator may be implemented as a K-1 stage convolutional filter with all unity weights as shown in figure 12b or by any of the equivalent recursive implementations shown in figures 12c, d, and e.

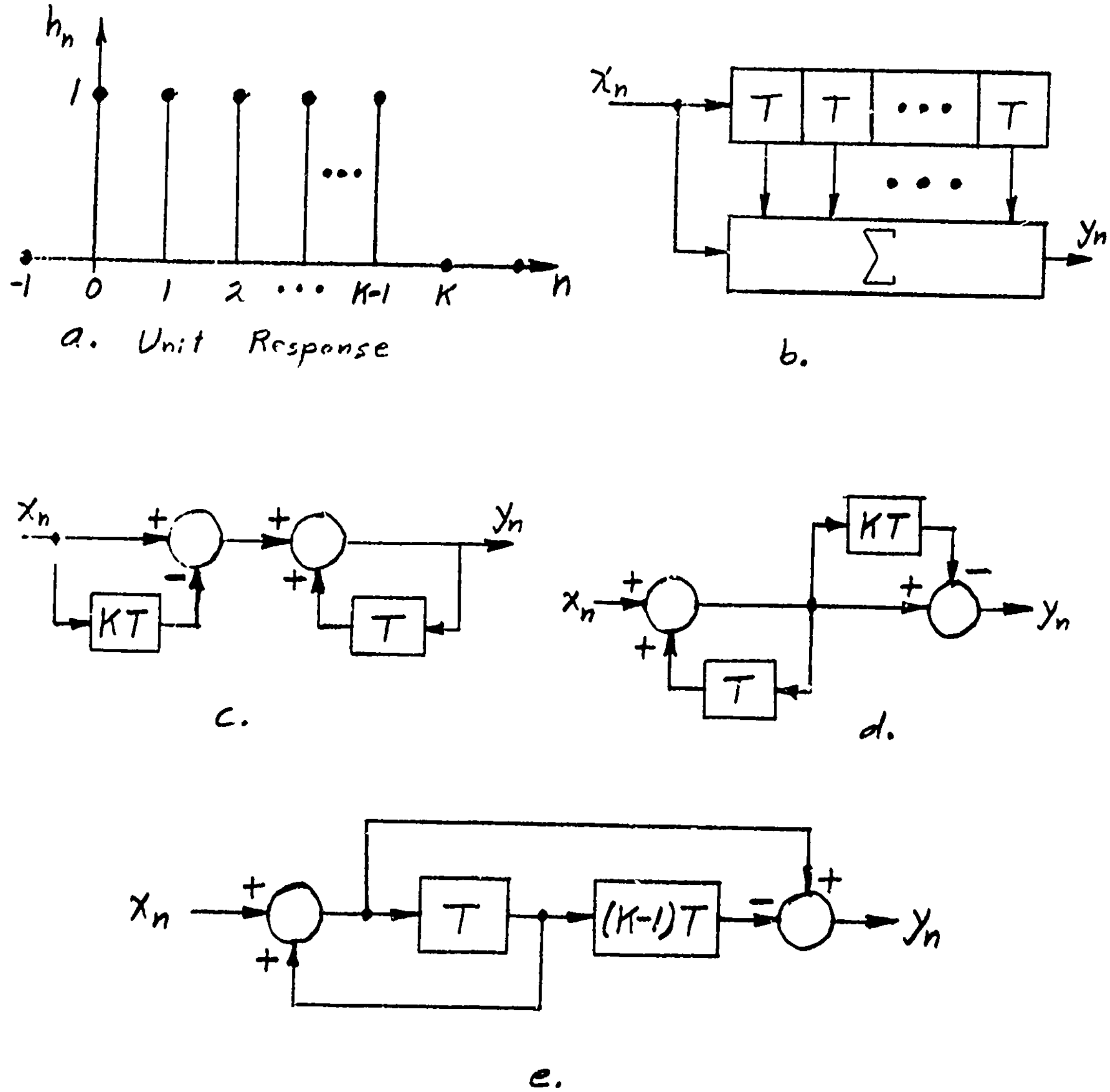


Figure 12: Store and Repeat Interpolation

To obtain the linear interpolation of figure 4c, but with a delay of KT seconds we require an interpolation filter with the unit response shown in figure 13a. This can be implemented as a 2K+1 stage convolutional filter as shown in figure 2a; with the tap weights given by figure 13a.

Since figure 13a can be obtained by convolving figure 12a with itself and introducing an additional delay of T seconds, the linear interpolator is implemented in figure 13b by cascading two store and repeat structures from figure 12c. Other realizations, are given in figures 13c and d. The last is obtained by noting that

$$H(z) = z^{-1}\left(\frac{1-z^{-K}}{1-z^{-1}}\right)^2 = \frac{z^{2K}-2z^{K+1}}{z^{2K+1}-2z^{2K}+z^{2K-1}}. \tag{15}$$

The structures in figures 13b, c, and d should include an attenuation factor, 1/K to be strictly consistent with figure 13a.

Now that we can realize interpolation filters, let us consider their frequency response. Since their unit response is symmetric about some value of time, their phase will vary linearly with frequency, resulting in zero differential delay in spite of the nonrecursive realizations. The frequency response of the store and repeat interpolator is obtained by taking the Fourier transform of

$$h(t) = \sum_{n=0}^{K-1} \delta(t-nT). \tag{16}$$

Thus

$$\mathcal{F}\left[h(t)\right] = \sum_{n=0}^{K-1} e^{-jn\omega T} = \sum_{n=0}^{K-1} (\cos n\omega T - j\sin n\omega T)$$

$$= \cos(K-1)\omega T/2\frac{\sin K\omega T/2}{\sin\omega T/2} - j\sin(K-1)\omega T/2\frac{\sin K\omega T/2}{\sin\omega T/2} \tag{17}$$

$$= \frac{\sin K\omega T/2}{\sin\omega T/2}e^{-j(K-1)\omega T/2}$$

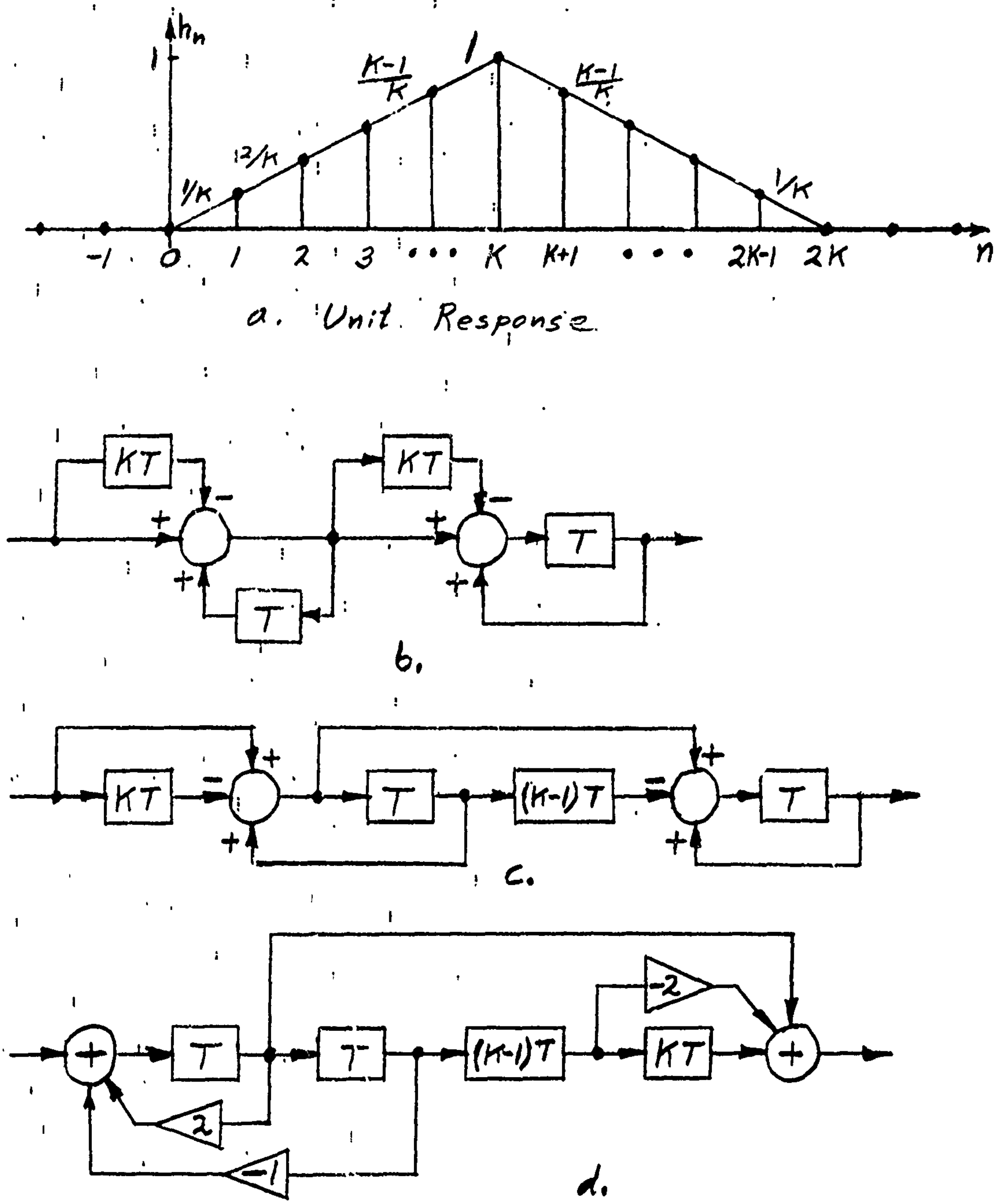


Figure 13: Linear Interpolation

where use has been made of series (465) and (466) on page 86 of reference 3. In terms of the frequency variable, $f = \omega/2\pi$,

$$H(f) = \frac{\sin K\pi ft}{\sin \pi fT} e^{-j(K-1)\pi fT}$$

$$= \frac{\sin \pi f/r}{\sin \pi f/Kr} e^{-j\frac{K-1}{K}\pi f/r} \tag{18}$$

where r was the sampling rate before interpolation. It is evident from the first line of the derivation that H(f) repeats with a period of $Kr = 1/T$:

$$H(f) = H(f+iKr). \tag{19}$$

It has zeroes at all multiples of r other than the iK^{th} (i.e., $H(ir) = 0$ for $i = 1, 2, \ldots, K-1$ and $H(Kr) = H(0) = K$). Furthermore,

$$\frac{1}{K} H(f) \longrightarrow \frac{\sin \pi f/r}{\pi f/r} \tag{20}$$

as $K \longrightarrow \infty$. For the case $K = 2$ we have

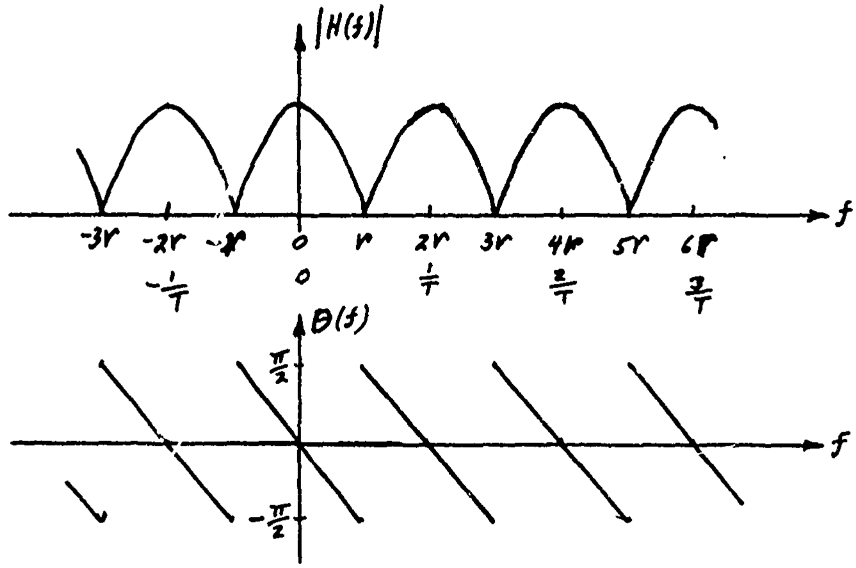
$$H(f) = \frac{\sin 2\pi fT}{\sin \pi fT} e^{-j\pi fT}$$

$$= 2\cos \pi fT e^{-j\pi fT} \tag{21}$$

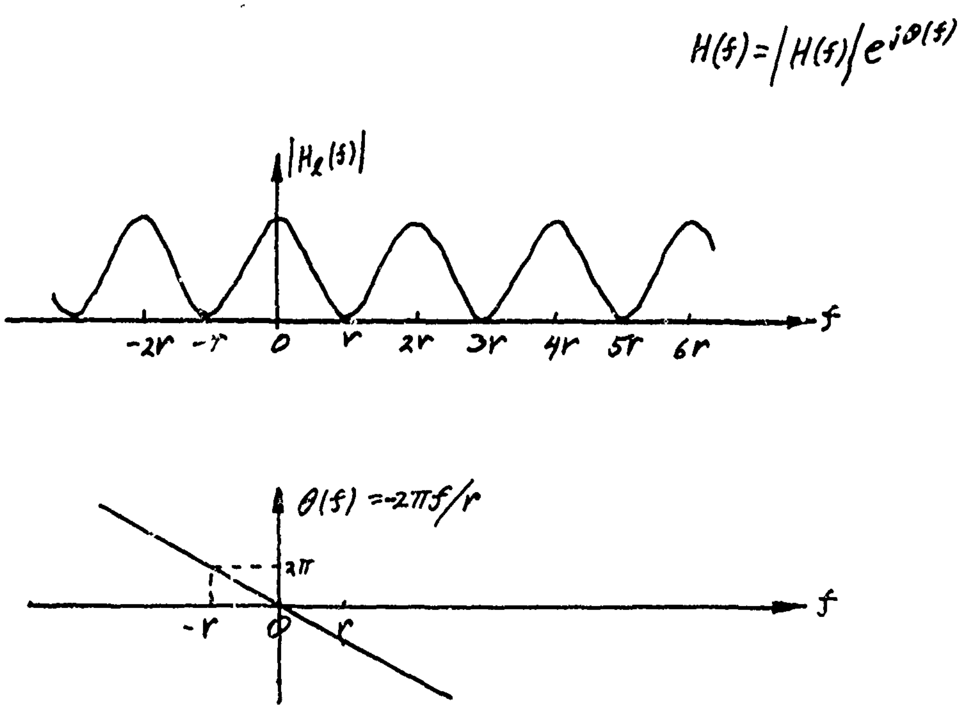
$$= 1 + e^{-j2\pi fT}$$

whose amplitude and phase spectra is plotted in figure 14a.

The frequency response of the linear interpolator is given by



a. Store and Repeat



b. Linear Interpolation

Figure 14: Amplitude and Phase Spectra for Two-to-One Interpolation

$$H_{\ell}(f) = \frac{1}{K}\left[H(f)\right]^2 e^{-j2\pi fT} = \frac{1}{K}\left(\frac{\sin K\pi fT}{\sin\pi fT}\right)^2 e^{-j2K\pi fT}$$

$$= \frac{1}{K}\left(\frac{\sin\pi f/r}{\sin\pi f/Kr}\right)^2 e^{-j2\pi f/r}. \tag{22}$$

This has period Kr, zeros at multiples of r other than the iK^{th}, and

$$\frac{1}{K} H_{\ell}(f) \longrightarrow \left(\frac{\sin\pi f}{\pi f}\right)^2 \tag{23}$$

as $K \longrightarrow \infty$. For the case K = 2,

$$H_{\ell}(f) = 2\cos^2 \pi fT e^{-j4\pi fT}$$

$$= (1+\cos 2\pi fT)e^{-j4\pi fT}, \tag{24}$$

as shown in figure 14b. Before interpolation the signal spectrum repeated every r hertz. Both interpolators in figure 14 serve to attenuate every other spectral repeat, with the linear interpolator doing a better job.

The analysis in Section III. 1. b of Reference 1 is valid only for large sampling ratios, K, since a $\frac{\sin\pi f/r}{\pi f/r}$ response is used, instead of $\frac{\sin\pi f/r}{\sin\pi f/Kr}$. Assuming K large and letting 2B represent the total signal bandwidth into the interpolator ($B \approx 1.5$KHz for SSB modulation and B = 3KHz for DSB and angle modulation) the ratio of the total energy contained in the unwanted spectral repeats to the desired energy was bounded in reference 1 for store-and-repeat interpolation:

$$ERH < 2\left(\frac{B}{r}\right)^2 \sum_{m=1}^{\infty} \frac{1}{\left(m-\frac{B}{r}\right)^2} \approx \frac{\pi 2}{3}\left(\frac{B}{r}\right)^2 \tag{25}$$

or

$$ERH < 5\text{-}20 \log \frac{r}{B} \qquad \text{db.} \tag{26}$$

Similary, for linear interpolation, we find:

$$\mathrm{ERH} < 2\left(\frac{B}{r}\right)^4 \sum_{m=1}^{\infty} \frac{1}{\left(m-\frac{B}{r}\right)^4} \approx \frac{\pi^4}{45}\left(\frac{B}{r}\right)^4 \tag{27}$$

or

$$\mathrm{ERH} < 3 - 40 \log \frac{r}{B} \quad \text{db.} \tag{28}$$

Thus, if the sampling rate before interpolation is $r = 16B$ (e.b., $B = 3\text{KHz}$ and $r = 48\text{KHz}$), then linear interpolation can produce more than 45 db attenuation of the unwanted energy. Since we chose the original sampling rate to be $r_1 = 8\text{KHz}$ we shall use the resampling filter to bring the rate up by a factor of 6, to $r_2 = 48\text{KHz}$.

Note that the sharp cut-off filter operates at the lowest possible rate, $r_1 = 8\text{KHz}$. The number of taps, N_1, which must be large because of stringent filter requirements, is minimized by using a low sampling rate. The multiplication rate for that filter is $N_1 r_1$ multiplications per second.

The resampling filter can have a relatively wide transition region between passband and stopband. It can thus operate at a higher sampling rate, $r_2 = 48\text{KHz}$, without an excessive number of taps, $N_2 \leq N_1$. Since its input is "zero filled" in transmitter operation and since only every 6^{th} output sample need be computed in receiver operation, the multiplication rate for this filter is only $N_2 r_1 (= N_2 r_2/6)$.

Finally since the interpolation filter will operate at an extremely high rate ($r_3 \approx 1\text{MHz}$), an implementation was found that requires no multiplications at all. This was possible because an extremely wide transition reqion is allowed. If the interpolation filter had required N_3 tap weights, its multiplication rate would have been $N_3 r_2$. Fortunately, an implementation was found for which $N_3 = 0$.

SECTION III

FINITE RESPONSE FILTER DESIGN

For a digital filter to have a perfectly linear phase versus frequency characteristic, its unit response must be symmetrical about some point:

$$h_n = \begin{cases} C_o & , n = M \\ \frac{1}{2}C_{|n-M|}, & |n-M| = 1, 2, \ldots, M \\ 0 & , |n-M| > M. \end{cases} \tag{1}$$

The combined requirements of symmetry ($h_{M-n} = h_{M+n}$) and physical realizability ($h_n = 0$ for $n<0$) imply a unit response of finite duration:

$$h_n = 0 \text{ for } n<0 \text{ and } n>2M.$$

Such a finite response can be readily implemented in a non-recursive (or convolutional) structure. The filter tap weights are simply the values of the unit response.

The frequency response corresponding to equation (1) is

$$\begin{aligned} H(f) &= \sum_{n=0}^{2M} h_n e^{-j2\pi nf/r} = e^{-j2\pi Mf/r} \sum_{K=0}^{M} C_K \cos(2\pi Kf/r) \\ &= e^{-j2\pi Mf/r} H_o(f). \end{aligned} \tag{2}$$

The problem is to choose the M values of C_K so that $H_o(f)$ adequately approximates a desired frequency response while the order of the filter, 2M+1, is kept small. There has been much recent activity and progress in the design of such finite response filters. This section will serve to summarize this progress and to introduce our design approach.

Finite response filters are commonly referred to as non-recursive (or transversal) filters. As we have seen with the interpolation filters discussed in section II. 4, finite response filters can be implemented recursively as well as non-recursively. A discussion of the relative merits of these two implementation approaches is deferred to Section VI. 1. Until then, we shall assume a non-recursive implementation for the filters discussed in this section.

1. WINDOW CARPENTRY

The classical design approach (4, 5, 6) has been to find the unit sample response corresponding to the desired frequency response and truncate it by an appropriate window function.

Consider the problem of approximating an ideal low-pass filter with zero phase, unity gain from -B to Bhz, and zero gain elsewhere:

$$H_o(f) = \begin{cases} 1, & -B<f<B \\ 0, & \text{elsewhere.} \end{cases} \tag{3}$$

The corresponding impulse response is

$$h(t) = 2B\frac{\sin 2\pi Bt}{2\pi Bt} .$$

The sampled data version of this filter has a unit response

$$\begin{aligned} h_n &= \frac{1}{r}\int_{-r/2}^{r/2} H_o(f)e^{j2\pi nf/r}df \\ &= 2\frac{B}{r}\,\frac{\sin\ 2\pi nB/r}{2\pi nB/r}\ , \quad n=0, \pm 1, \pm 2, \ldots \end{aligned} \tag{4}$$

and an ideal filter spectrum that repeats at the sampling rate, r (r>2B). This ideal filter, however, requires a unit response of infinite duration. Simple truncation to 2M+1 terms by merely setting $h_n = 0$ for $|n|>M$ (the so called rectangular window) results in a 9 percent frequency response overshoot at the band edges (the so called Gibbs phenomenon [7]).

Multiplying h_n by a finite duration window function w_n (w_n = 0 fon $|n|>M$) results in a finite duration unit response whose spectrum is the convolution of $H_o(f)$ with $W(f)$ the Fourier transform of the window function. The window function is chosen so that $W(f)$ has a narrow main lobe with small sidelobes. The rectangular window

$$w_n = \begin{cases} 1, & n=0, \pm 1, \pm 2, \ldots, \pm M \\ 0, & \text{elsewhere} \end{cases} \tag{5}$$

is inadequate because of the sidelobes in its Fourier transform. Triangular and truncated gaussian windows offer some improvement, though decreased sidelobes come at the expense of a widened main lobe (for fixed M). Windows of the form

$$w_n = C+(1-C)\cos(\pi n/M), \quad n=0, \pm 1, \pm 2, \ldots, \pm M \tag{6}$$

have historically(4) been found useful, in that they serve to cancel the first $(\sin x)/x$ sidelobe of the rectangular window without appreciably widening the main lobe. For C = 0.54 we have a Hamming window, for C = 0.5 we have a Hanning window, and for C = 0.56 we have Stockham's(8) modified Hamming window. A variety of other more complex, windows (including the Kaiser and Dolph-Chebyschev windows) have been found even more effective.(5, 9)

The major advantage of the time-domain window approach is the extreme simplicity of the filter design procedure.

2. FREQUENCY SAMPLE SPECIFICATION

This approach, introduced by Gold and Jordan(10) and developed by Rabiner, Gold and McGonegal,(11, 12) specifies the filter response at discrete frequencies.

Consider the ideal low-pass filter, $H_o(f)$, which is an even function of frequency. The frequency range, 0 to $\frac{r}{2}$, is divided into M segments and

$$H_i = H\left(\frac{ir}{2M}\right), \quad i=0, 1, 2, \ldots, M \tag{7}$$

is specified to be unity for $\frac{ir}{2M}<B$ and zero for $\frac{ir}{2M}>B$, with $H_{-i} = H_i$. These values can be viewed as the Fourier Series coefficients of a periodic version of h(t). Instead of being truncated, h(t) is aliased by its periodicity. The sampled version of this h(t) furnishes a possible set of filter tap weights, h_n. The weights h_n can be obtained from $H(\frac{ir}{2M})$ by the Discrete Fourier Transform (utilizing an FFT algorithm). Unfortunately, the corresponding spectrum, while agreeing with the ideal at the specified points, is far from ideal elsewhere. To overcome this problem, a few frequency samples in the neighborhood of the transition frequency ($\pm$B hertz) are allowed to vary in amplitude. The overall frequency response is thus improved (as in the window approach) by reducing the ripple near the band edges at the expense of broadening the transition band.

This procedure usually outperforms the window approach, but at the expense of a linear search for the optimum values of the frequency samples being varied.

3. ZERO PLACEMENT

The previous approach specifies a uniform placement of real frequency zeros in the stopband. A variant due to Requicha and Voelcker(13) specifies the non-uniform placement of all real and complex zeros, resulting in filter stopbands that are specified in terms of attenuation rates measured in decibles per octave.

If N_v real frequency zeros are uniformly spaced over the entire r hertz frequency range and M of these are replaced by N_p complex passband zeros (leaving $N_s = N_v - M$ stopband zeros) an approximate asymptotic attenuation rate of

$$A = -6(M-N_p) = -6\left[N_v-(N_s+N_p)\right] \quad \text{db/octave} \tag{8}$$

is achieved. Thus, high attenuation rates are achievable by closely packing the stopband zeros.

4. EQUAL RIPPLE SPECIFICATION

The most sophisticated approach to nonrecursive filter design is to specify the order of the filter and the maximum allowable ripple in both the passband and stopband. A non-linear optimization technique is then used to specify the filter meeting these contraints with a minimum transition band.(14) Herrmann has available the tap weights of 400 filters designed in this manner. An efficient algorithm (due to Hofstetter(15)) for the design of

optimum non-recursive digital filters having prescribed equal stopband and equal passband ripples, is outlined below.

The filter specification is given in terms of the maximum allowable deviation from unity in the passband, δ_p, the maximum allowable deviation from zero in the stop band, δ_s, the maximum number of extrema in the passband, n_p, and the maximum number of extrema in the stopband, n_s. The filter of minimum order, 2M+1, will have $M = n_p+n_s+1$. The passband edge, f_p, and the stopband edge, f_s, are defined as the frequencies at which the response first leaves and finally enters the corresponding tolerance region, $|H(f)-1| \leq \delta_p$ and $|H(f)| \leq \delta_s$. The narrowest transition band, $|f_s-f_p|$, occurs when the filter is equiripple in both the passband and the stopband. Since

$$H_o(f) = \sum_{K=0}^{M} C_k \cos 2\pi K f/r \tag{9}$$

can be written as

$$H_o(f) = \sum_{K=0}^{M} d_K x^K, \tag{10}$$

where $x = \cos 2\pi f/r$, a trigonometric polynomial approximation to the desired frequency characteristic is being sought.

The design algorithm begins with an initial set of M+1 frequencies ($0 = f_o<f_1<f_2<\ldots<f_m = r/2$) and a Lagrange interpolation polynomial that goes through the values $1 \pm \delta_p$ and $\pm\ \delta_s$ at these frequencies (the black dots in figure 15). The second iteration utilizes the frequencies at which the extrema of the first polynomial occur (the open dots in figure 15). The procedure, utilizing a varient of the barycentric form of the Lagrange interpolaration formula, is continued until all the extremum frequencies are equal to their counterparts from the previous iteration. The filter tap weights, h_n, are obtained from this final H(f) as a discrete fourier transform of $H(\frac{ir}{2M})$.

Slight variations of this procedure can be used to design bandpass filters, low pass differentiators, etc. For example, a bandpass differentiator having an ideal frequency response

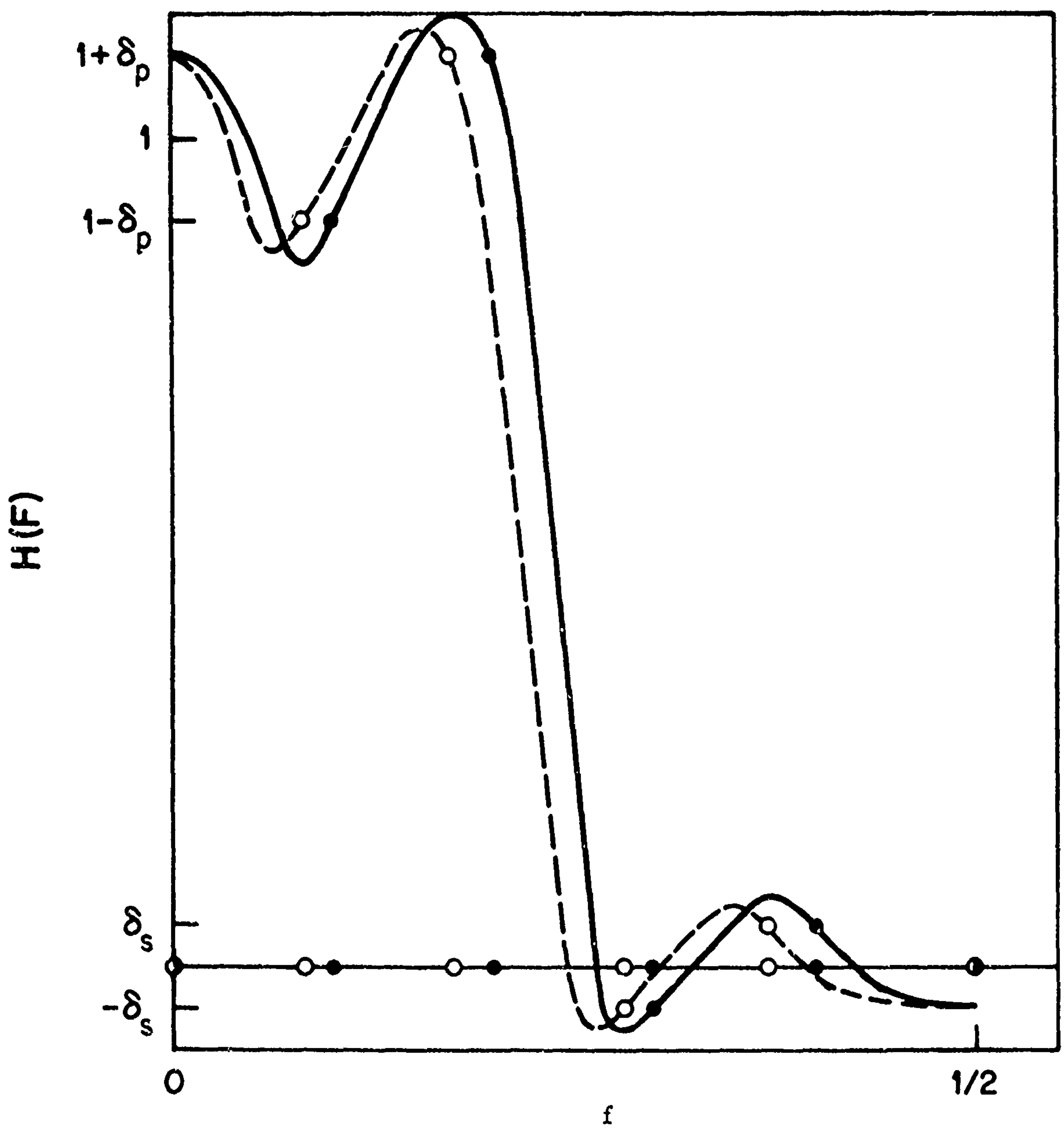


Figure 15: The Hofstetter Finite-Response Design Algorithm for an 11-tap Low-Pass Filter with $n_p = n_s = 2$

$$H(f) = \begin{cases} 0 \quad , & |f|<b \\ j2\pi f, & b<|f|<B \\ 0 \quad , & |f|>B \end{cases} \tag{11}$$

would be specified to be purely imaginary and to remain within $\pm\,\delta_1$ and $\pm\,\delta_2$ in the two stopbands and within $2\pi f(1\pm\,\delta_p)$ in the passband (i.e., the percentage error in the passband is specified). The fact that the procedure converges rapidly and yields an optimum design makes it extremely attractive.

5. QUANTIZATION EFFECTS

Errors are introduced in convolutional filters through limited word-lengths for the state variables (the signal values stored in the shift registers) and for the filter tap weights (the constants stored in read-only-memories). Errors of the first type, including the rounding of the result of the convolution, can be described as additive noise with variance $E^2/12$, where E is the quantization level difference. Errors of the second kind, however, can alter the frequency response of the filter.

A filter designed according to one of the preceeding methods to satisfy certain ripple specifications, will no longer meet those specifications when the tap weights are rounded. However there may exist a set of weights, within the word length limitation, but different than the rounded version of the ideal weights, that does satisfy the filter requirements. Avenhaus and Schussler[16, 17] have proposed a modified Gauss-Seidel procedure for searching the discrete parameter space in the neighborhood of the ideal, for an optimum set of quantized tap weights.

6. INTERACTIVE DESIGN AND THE RAISED-COSINE ROLL-OFF

As pointed out in the last section, most algorithms for optimum filter design are relatively meaningless if the filter tap weights are rounded to accomodate a word-length limitation. Our approach for an expedient (rather than optimum) design has been a human search on a computer terminal. A computer program was written, based on the simple window approach and designed for man-machine interaction. The program rounds the tap weights and performs a fast Fourier transform on these rounded weights to furnish the actual frequency response. The number of bits precision (word-length), L, the window parameter, C, the 6 db cutoff point, B, and M (where 2M+1 is the nominal number of filter taps) are parameters at the disposal of the designer. Because the magnitude of

the tap weights decrease as n increases, the weights eventually fall below $E/2$ where E is the quantization level difference. These weights are rounded to zero, so that the actual order of the filter, N, is less than $2M+1$.

Specifications were set on the allowable passband and stopband ripples and on the maximum allowable transition width. A search for the filter of minimum order for a given precision revealed that the minimum number of taps occurred at the values of the parameters C and M for which the natural truncation phenomenon was most pronounced, i.e., when $(2M+1)-N$ was a maximum.

This led to consideration of the raised-cosine filter characteristic.[18] The ideal filter characteristic

$$H_o(f) = 1, \quad 0 \leq f \leq B \tag{12}$$

shown in figure 16a corresponds to an impulse response:

$$h(t) = 2B\frac{\sin(2\pi Bt)}{2\pi Bt} \tag{13}$$

which decays asymptotically as $1/t$ and has a corresponding unit response:

$$h_n = \frac{\sin(2\pi nB/r)}{\pi n} . \tag{14}$$

The raised-cosine filter characteristic with roll-off α

$$H_1(f) = \begin{cases} 1, & 0 \leq f \leq (1-\alpha)B \\ 1/2\left[1 - \sin\frac{\pi(f-B)}{2\alpha B}\right], & (1-\alpha)B \leq f \leq (1+\alpha)B \end{cases} \tag{15}$$

shown in figure 16b corresponds to an impulse response

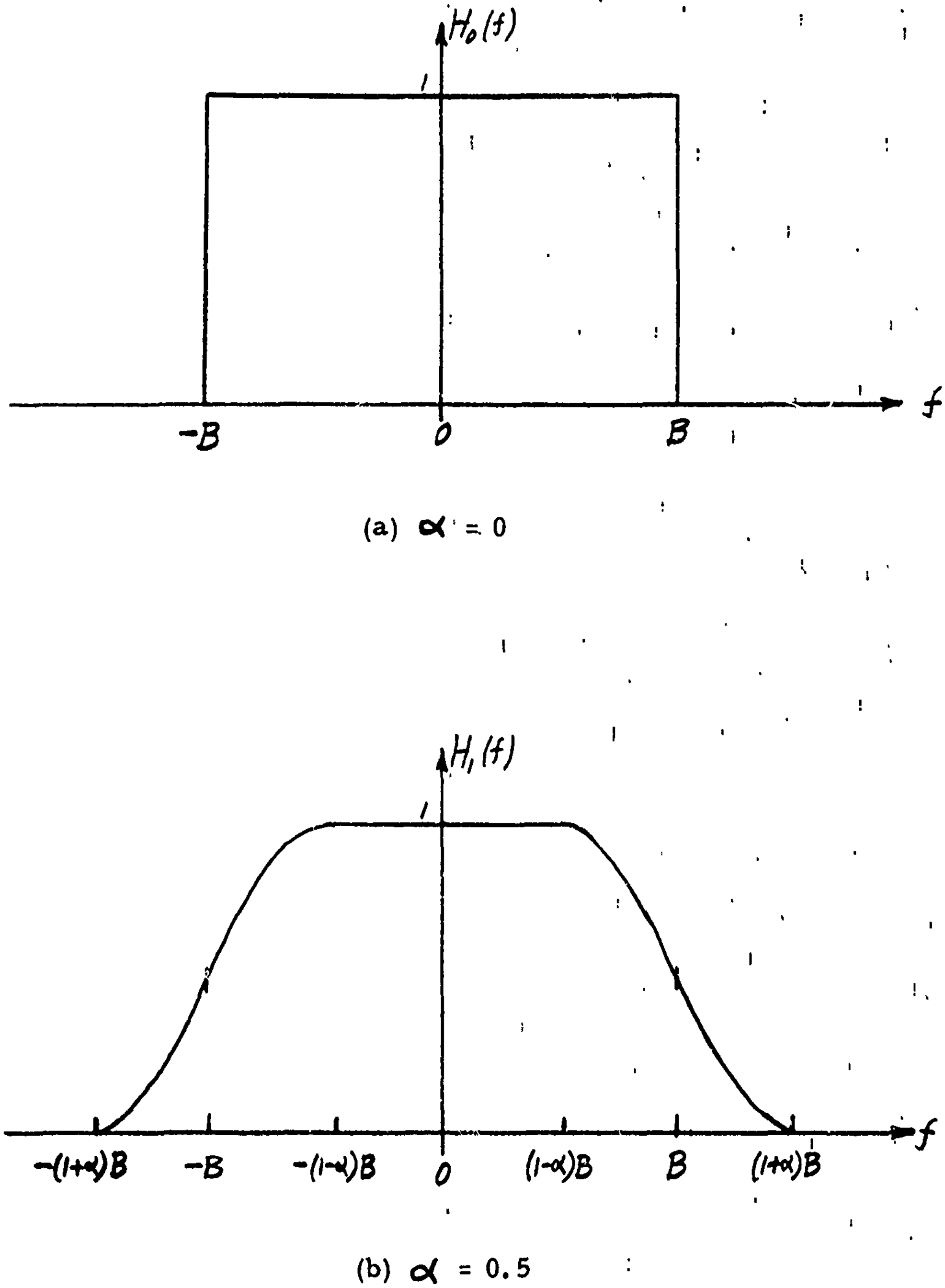


Figure 16: The Raised-Cosine Filter Characteristic

$$h(t) = 2B\frac{\sin(2\pi Bt)}{2\pi Bt} \quad \frac{\cos(2\pi\alpha Bt)}{1-4(2\alpha Bt)^2} \tag{16}$$

which decays as $1/t^3$ for large t. The corresponding unit response,

$$h_n = \frac{\sin(2\pi nB/r)}{\pi n} \quad \frac{\cos(2\pi\alpha nB/r)}{1-4(2\alpha nB/r)^2} \tag{17}$$

with the second factor $\cos(\pi x)/(1-4x^2)$ set equal to $\pi/4$ when $x = 1/2$, will enhance the natural truncation phenomenon because of the $1/n^3$ decay. By utilizing a small value for α, a filter of low order N is obtained without too great a sacrifice in transition width ($f_s - f_p \approx 2\alpha$). Unfortunately, this natural truncation gives rise to ripples in much the same way as does rectangular truncation. However, when this technique is used in conjunction with a window,

$$h_n = \frac{\sin(2\pi nB/r)}{\pi n} \cdot \frac{\cos(2\pi\alpha nB/r)}{1-4(2\alpha nB/r)^2} \cdot \left[C+(1-C)\cos(\pi n/M)\right], \tag{18}$$

excellent results are achieved.

The details of the interactive filter design program are given in Section VI. 2.

SECTION IV

BANDPASS SAMPLING

1. IN-PHASE AND QUADRATURE SAMPLING

In this section we will derive some of the basic mathematical relations pertaining to the in-phase and quadratrue sampling technique used for translating a bandpass signal to lowpass. In general we will be concerned with a real signal s(t) whose Fourier transform S(ω) is band limited and centered approximately at $\pm f_c$. More specifically let

$$F\left[s(t)\right] = S(\omega), \text{ where } S(\omega)=0 \text{ for } f_c-\frac{W}{2}\geq|f|\geq f_c+\frac{W}{2} \tag{1}$$

as shown in figure 17a.

The analytic signal Z(t) of s(t) is defined as

$$Z(t) = s(t)+j\hat{s}(t) \tag{2}$$

where $\hat{s}(t)$ is the Hilbert Transform of s(t). The Fourier Transforms of Z(t) and $\hat{s}(t)$ are respectively given by

$$F\left[Z(t)\right] = \begin{cases} 2S(\omega) & \text{for } \omega\geq 0 \\ 0 & \text{for } \omega<0 \end{cases} \tag{3}$$

$$F\left[\hat{s}(t)\right] = \begin{cases} -jS(\omega) \text{ for } \omega\geq 0 \\ jS(\omega) \text{ for } \omega<0 \end{cases} \tag{4}$$

In words, the spectrum of the analytic signal Z(t) is the one sided spectrum of s(t) and the spectrum of the Hilbert Transform $\hat{s}(t)$ is the spectrum of s(t) with the negative frequencies shifted $\pi/2$ and the positive frequencies shifted $-\pi/2$.

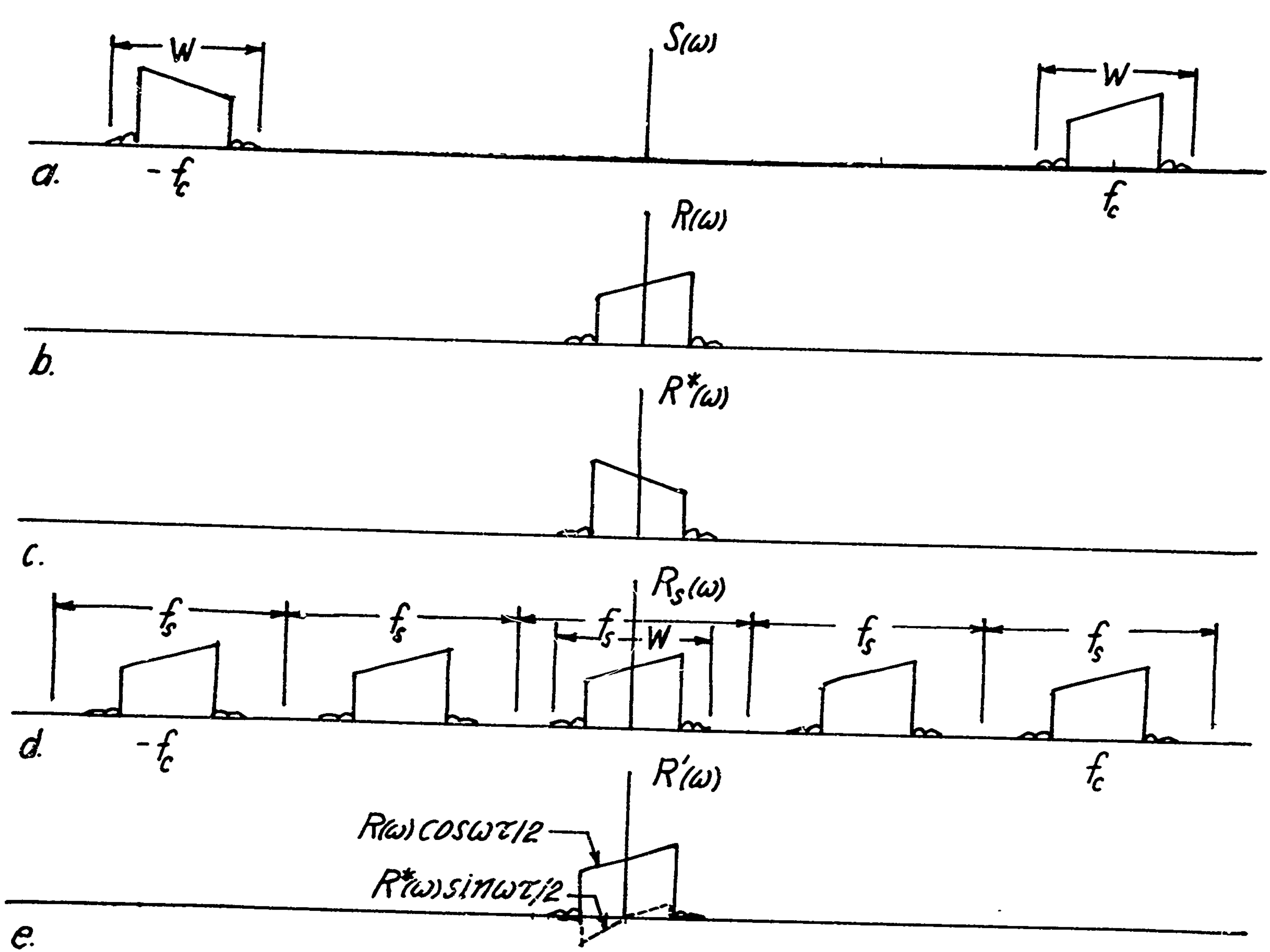


Figure 17: Waveform Spectra for In-Phase and Quadrature Sampling

Let us now define R(t) the complex envelope of s(t) at frequency f_c by the equation

$$R(t) = Z(t)\exp\left[-j\omega_c t\right] = I(t)+jQ(t) \tag{5}$$

where I(t) and Q(t) can be seen from definition (5) to be given by

$$\begin{aligned} I(t) &= s(t)\cos\omega_c t+\hat{s}(t)\sin\omega_c t \\ Q(t) &= \hat{s}(t)\cos\omega_c t-s(t)\sin\omega_c t \end{aligned} \tag{6}$$

Also from equations (6) we can solve for s(t) and $\hat{s}(t)$ to obtain

$$\begin{aligned} s(t) &= I(t)\cos\omega_c t-Q(t)\sin\omega_c t \\ \hat{s}(t) &= I(t)\sin\omega_c t+Q(t)\cos\omega_c t \end{aligned} \tag{7}$$

The Fourier Transform of R(t) denoted by R(ω) is the transform of the analytic signal Z(t) translated down in frequency by f_c. Similarly the spectrum R*(ω) of the complex conjugate R*(t) of R(t) is the spectrum the conjugate Z*(t) of the analytic signal translated up in frequency by f_c, i.e. the negative frequency part of S(ω) translated up by f_c. Both R(ω) and R*(ω) are shown in figures 17b and 17c respectively. Note that R*(ω) is not the complex conjugate of R(ω). If we denote the complex conjugate of R(ω) by $\overline{R(\omega)}$ then R*(ω) is given by

$$R^*(\omega) = \overline{R(-\omega)} = A_r(-\omega)-jB_r(-\omega)$$

where

$$R(\omega) = A_r(\omega)+jB_r(\omega) \tag{8}$$

and $A_r(\omega)$ and $B_r(\omega)$ are the real and imaginary components of the spectrum R(ω).

Now let I(ω) and Q(ω) respectively represent the Fourier Transform of I(t) and Q(t). From the definitions of R(ω) and R*(ω) we thus leave

$$A_r(\omega)+jB_r(\omega) = I(\omega)+jQ(\omega)$$

and (9)

$$A_r(-\omega)-jB_r(-\omega) = I(\omega)-jQ(\omega)$$

from which we obtain

$$I(\omega) = 1/2\left[A_r(\omega)+A_r(-\omega)\right] +j1/2\left[B_r(\omega)-B_r(-\omega)\right]$$

and (10)

$$Q(\omega) = 1/2\left[B_r(\omega)+B_r(-\omega)\right] -j1/2\left[A_r(\omega)-A_r(-\omega)\right]$$

Consider now the case where we sample the real bandpass signal s(t) at times

$$t_{mi} = mT$$

and

$$t_{mq} = mT-\tau$$

where (11)

$$T = nT_c \leq 1/W \text{ and } T_c = 1/f_c$$

and

$$\tau = kT_c+T_c/4.$$

and where m, n and k are integers.

In words, we sample s(t) at times t_{mi} to obtain the in-phase samples and at times t_{mq} to obtain the quadrature samples. The period T as noted from equations (11) contains an integral number of cycles at frequency f_c. Thus the mth in-phase sample is given by

$$s(t_{mi}) = I(mT) \tag{12}$$

which is obtained from equation (7) after replacing t with t_{mi}.

Similarly the quadrature samples are obtained at times t_{mq} which differ from the in-phase sampling times by one quarter cycle of frequency f_c. Thus the mth quadrature sample is similarly obtained by substituting

t_{mq} in equation (7) i.e.

$$S(t_{mq}) = Q(mT-\tau) \tag{13}$$

We will now show that the in-phase and quadrature sampling technique can recover to a good approximation the spectrum $R(\omega)$ of the complex envelope. Inasmuch as $R(\omega)$ contains all of the information in $s(t)$. Recovery of $R(\omega)$ would therefore be adequate. Since the signal is sampled at a rate f_s we will in fact not recover $R(\omega)$ but rather $R_s(\omega)$ which is $R(\omega)$ repeated periodically with period f_s. If $f_s \geq W$ there will be no spectral overlaps and therefore $R_s(\omega)$ as shown in figure 17d will also contain all of the information in the original $R(\omega)$.

Consider now the two functions $I(t)$ and $Q(t-\tau)$ the sampled values of these two continuous functions being our in-phase and quadrature samples given by equations (12) and (13). The Fourier Transform of these functions are obviously given by

$$F\Big[I(t)\Big] = I(\omega)$$

and

$$F\Big[Q(t-\tau)\Big] = Q(\omega)e^{-j\omega\tau} \tag{14}$$

and therefore the recovered spectrum $R'(\omega)$ for the continuous case can be written as

$$R'(\omega) = I(\omega)+jQ(\omega)e^{-j\omega\tau}$$

which after some algebraic manipulations and using equations (8), (9) and (10) becomes

$$R'(\omega) = \Big[R(\omega)\cos\omega\tau/2+jR^*(\omega)\sin\omega\tau/2\Big]e^{-j\omega\tau/2} \tag{15}$$

The recovered spectrum $R'_s(\omega)$ of the sampled waveform will be the same as in equation (15) repeated periodically with period equal to the sampling rate f_s. The recovered spectrum after in-phase and quadrature sampling will thus consist of the desired spectrum $R(\omega)$ filtered by a $\cos\omega\tau/2$ term and an undesired term $R^*(\omega)$, (the negative frequencies term of the original

spectrum $S(\omega)$), filtered by $j\sin\omega\tau/2$ as shown in figure 17e. The $\exp(-j\omega\tau/2)$ term of course represents a time delay of $\tau/2$ and is of no significant consequence. Clearly from equation (15) one would want to make τ as small as possible in order to minimize the effects of the interfering term $R^*(\omega)$. From equation (11) it can be seen that the smallest τ can be made in $T_c/4$ which would result in filter characteristics of $\cos\omega T_c/8$ and $\sin\omega T_c/8$ respectively for the desired and undesired terms. The extent to which the recovered signal thus described will differ from the desired signal is thus seen clearly to depend on the ratio W/f_c. Thus if $W/f_c \ll 1$, i.e. if the narrowband approximation holds, then the interfering term would be negligible otherwise it will not. If the narrowband approximation does not hold or if the requirements are such that the interfering term cannot be tolerated then one must find ways of eliminating or reducing this interference. We will describe two approaches which will enable us to reduce the interfering component $R^*(\omega)\sin\omega\tau/2$. In the first approach the sampled values of $I(t)$ and $Q(t-\tau)$ are filtered digitally while in the second approach instead of sampling twice per period T we sample more often to obtain sampled values of $I(t)$, $I(t\underline{+}2\tau)$, $I(t\underline{+}4\tau)$... and $Q(t\underline{+}\tau)$, $Q(t\underline{+}3\tau)$...

2. FILTERING I(t) and Q(t-τ)

Consider a pair of conjugate filters F and f* given by

$$F = |F(\omega)| \exp\left[j\phi(\omega)\right]$$

and

$$F^* = |F(\omega)| \exp\left[-j\phi(\omega)\right], \tag{16}$$

and let us filter the $I(t)$ and $Q(t-\tau)$ recovered components of the signal respectively with the above two filters. Our recovered signal spectrum $R'(\omega)$ will thus be given by

$$R'(\omega) = I(\omega)\,|F(\omega)|\exp\left[j\phi(\omega)\right] + jQ(\omega)\,|F(\omega)|\exp\left[-j\phi(\omega)-\omega\tau\right]. \tag{17}$$

Again after some algebraic manipulations and using equations (8), (9) and (10) equation (17) becomes

$$R'(\omega) = \Big\{R(\omega)\,\big|F(\omega)\big|\cos\big[\phi(\omega)+\omega\tau/2\big] + jR^*(\omega)\,\big|F(\omega)\big|\sin\big[\phi(\omega)+\omega\tau/2\big]\Big\}\exp(-j\omega\tau/2). \tag{18}$$

As in equation (15) the $\exp(-j\omega\tau/2)$ term represents only a time delay and is of no significant consequence.

If the recovered signed $R'(\omega)$ is not to contain any interference terms then the magnitude $|F(\omega)|$ of the filters F and F* must equal to unity and the phase $\phi(\omega)$ must equal $-\omega\tau/2$, i. e. the phase must be linear. The impulse response of the F and F* filters will thus be a $(\sin x)/x$ delayed (or advanced) by $\tau/2$ with zero crossings at intervals T. Figure 18 shows a convolutional digital filter implementation of these filters where each filter has 2n+1 tap weights and the tap weights a_k are given by

$$a_k = (-1)^k (\sin\omega_s\tau/4)/(k\pi+\omega_s\tau/4). \tag{19}$$

$$k = 0, \pm 1, \ldots \pm n$$

If the narrowband approximation is relatively good i. e. if $W/f_c \ll 1$ then a very small number of filter taps will be required to give a very good approximation of $\phi(\omega)$ to $-\omega\tau/2$. For ten to twenty percent bandwidths five to seven tap filters would be all that would be required for any reasonable amount of undesired signal rejection.

3. THE MULTIPLE SAMPLE APPROACH

Let the bandpass signal s(t) be sampled more than twice (as for in-phase and quadrature sampling) during each sampling period $T = 1/W$ and let the intervals between multiple samples be multiples of $\tau = T_c/4$, i. e. let us sample s(t) and times

$$mT,\ mT-\tau,\ mT+\tau,\ mT-2\tau,\ mT+2\tau, \ldots \tag{20}$$

where as before

$$\tau = T_c/4 \text{ and } T = nT_c \le 1/W \tag{21}$$

and m and n are integers.

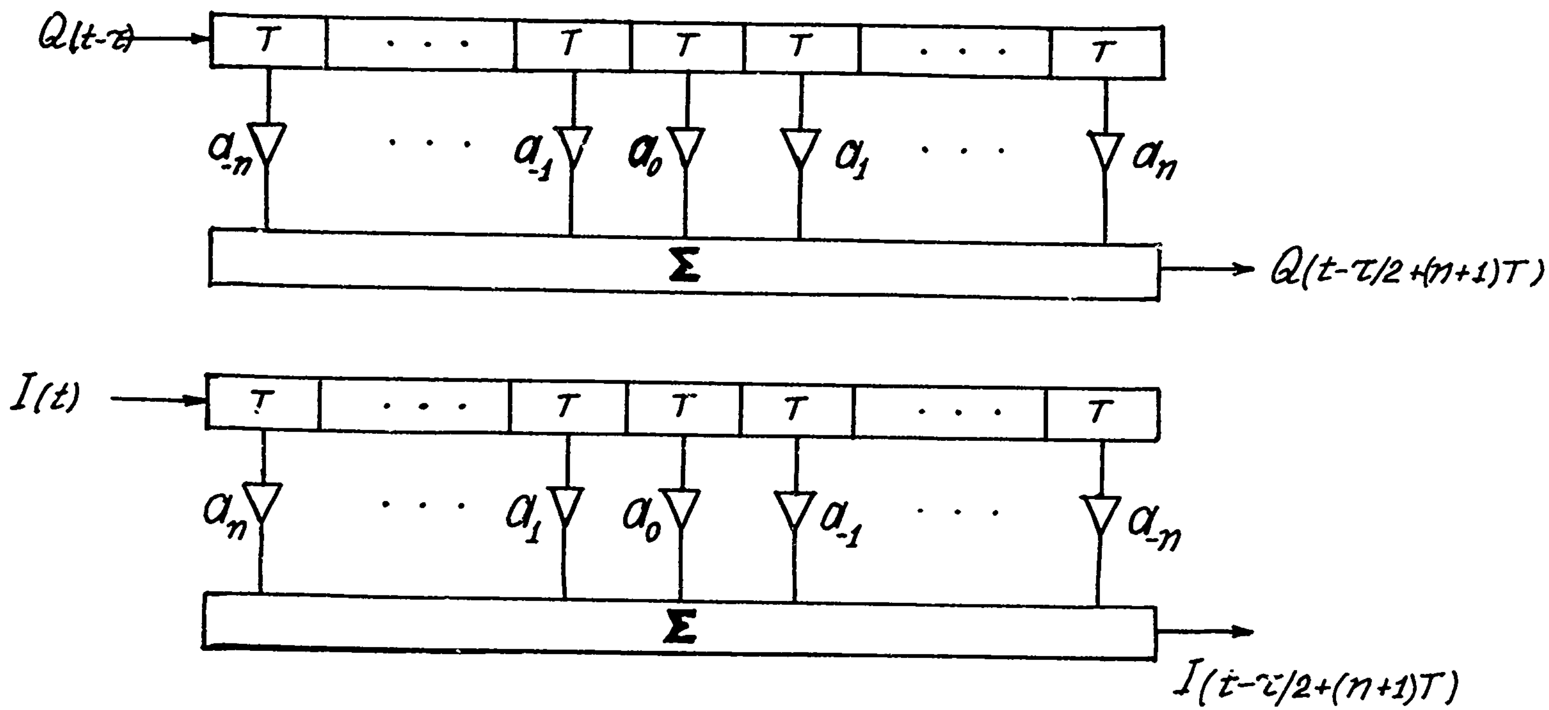


Figure 18: Filtering of the In-Phase and Quadrature Sampled Waveforms

From equation (7) it is clear that for the sample times above we would obtain sampled values of I(t) and Q(t) given by

$$I(t),\ Q(t-\tau),\ -Q(t+\tau),\ -I(t-2\tau),\ -I(t+2\tau),\ldots \tag{22}$$

where mT is replaced by t.

We have shown (equation (15)) that if the first two samples from (22) are used the recovered signal spectrum $R'(\omega)$ consists of the desired spectrum $R(\omega)$ multiplied by $\cos\omega\tau/2$ and the undesired spectrum $R^*(\omega)$ multiplied by $\sin\omega\tau/2$. It has been shown[19] that in general if n+1 samples are used as given by (22) then the recovered signal spectrum $R'(\omega)$ will consist of the desired spectrum $R(\omega)$ multiplied by $(\cos\omega\tau/2)^n$ and the undesired spectrum $R^*(\omega)$ multiplied by $(\sin\omega\tau/2)^n$, provided the samples are combined appropriately, namely weighted by the binomial coefficients.

To show that this is as we will derive the expressions for $R'(\omega)$ for n = 2, 3, and 4. From equations (8) and (9) we see that

$$I(\omega) = R(\omega)+R^*(\omega)$$

and

$$jQ(\omega) = R(\omega)-R^*(\omega).$$

If we weigh the first three terms of expression (22) by 1, j/2 and -j/2 and add we obtain

$$\begin{aligned} R'(\omega) &= R(\omega)+R^*(\omega)+\Big[R(\omega)-R^*(\omega)\Big]\exp(-j\omega\tau)/2 \\ &\qquad +\Big[R(\omega)-R^*(\omega)\Big]\exp(j\omega\tau)/2 \\ &= R(\omega)(1+\cos\omega\tau)+R^*(\omega)(1-\cos\omega\tau) \qquad (23) \\ &= 2R(\omega)(\cos\omega\tau/2)^2+2R^*(\omega)(\sin\omega\tau/2)^2. \end{aligned}$$

For the case n=3 we weigh the samples with the weights 3, j3, -j, -1 and add. Thus we obtain

$$
\begin{aligned}
R'(\omega)\exp(j\omega\tau/2) &= 3\left[R(\omega)+R^*(\omega)\right]\exp(j\omega\tau/2) \\
&\quad +3\left[R(\omega)-R^*(\omega)\right]\exp(-j\omega\tau/2)+\left[R(\omega)-R^*(\omega)\right]\exp(j3\omega\tau/2) \\
&\quad +\left[R(\omega)+R^*(\omega)\right]\exp(-j3\omega\tau/2) \qquad (24) \\
&= 8R(\omega)(\cos\omega\tau/2)^3+j8R^*(\omega)(\sin\omega\tau/2)^3
\end{aligned}
$$

Similarly for n=4 the weights would be 6, +j4, -j4, -1, and -1. Thus we have

$$
\begin{aligned}
R'(\omega) &= 6\left[R(\omega)+R^*(\omega)\right]+4\left[R(\omega)-R^*(\omega)\right]\exp(-j\omega\tau) \\
&\quad +4\left[R(\omega)-R^*(\omega)\right]\exp(j\omega\tau)+\left[R(\omega)+R^*(\omega)\right]\exp(-j2\omega\tau) \qquad (25) \\
&\quad +\left[R(\omega)+R^*(\omega)\right]\exp(j2\omega\tau) \\
&= 16R(\omega)(\cos\omega\tau/2)^4+16R^*(\omega)(\sin\omega\tau/2)^4.
\end{aligned}
$$

We thus see that by appropriately combining the various samples the filter functions $(\cos\omega\tau/2)^n$ and $(\sin\omega\tau/2)^n$ can be synthesized to filter $R(\omega)$ and $R^*(\omega)$ respectively. By selecting other than the binomial coefficients to weigh the various samples other filter functions can be synthesized. For example for N=2 if the weights are 1, jk/2 and -jk/2 we can obtain

$$
R'(\omega) = R(\omega)\left[1+k\cos\omega\tau\right]+R^*(\omega)\left[1-k\cos\omega\tau\right]. \qquad (26)
$$

For n=3 and weights 3, j3, -jk, and -k we obtain

$$
\begin{aligned}
R'(\omega) &= R(\omega)\left[8k(\cos\omega\tau/2)^3+6(1-k)\cos\omega\tau/2\right] \\
&\quad +jR^*(\omega)\left[8k(\sin\omega\tau/2)^3+6(1-k)\sin\omega\tau/2\right] \qquad (27)
\end{aligned}
$$

Thus we see that instead of obtaining an nth order zero at the origin as would be the case for $(\sin\omega\tau/2)^n$ one can obtain n evenly spaced zeros by proper choice of the coefficient k as in the case above or by proper choice of additional coefficients for higher order n.

4. COMPARISON BETWEEN FILTERING AND MULTIPLE SAMPLING

As was shown by the description of the two techniques one can achieve very effective rejection of the unwanted interfering signal $R^*(\omega)$ by either filtering the sampled I and Q waveforms, or by a suitable combination of a multiplicity of n+1 samples. In any specific application the choice which if any of these techniques is preferable will depend on the complexity of implementation. In general one will perform the analog to digital conversion either by a single device capable of operating at a rate of $4f_c$ or a multiplicity of devices each capable of operating at a rate $f_s = W$. The cost of analog to digital converters increases quite rapidly with increasing sampling rate. Therefore if the bandpass signal is a relatively wideband signal where the rates $4f_c$ and f_s not significantly different then a single analog to digital converter would be preferred and as many samples as possible would be obtained to achieve maximum interference rejection. If on the other hand the ratio of $4f_c$ to f_s is large then it would be advantageous to leave two analog to digital converters operating at a rate f_s with the sampled in-phase and quadrature signals then being filtered to achieve the required filtering. For the purposes of the transceiver presently under consideration the receiver IF will be in the neighborhood of 100 to 200 kHz and the IF signal bandwidths will be from 4 to 25 kHz. In view of these considerations the in-phase and quadrature sampling followed by a relatively minor pair of filters is clearly indicated as the optimum approach.

SECTION V

MODULATION AND DEMODULATION

1. MODULATION AND FREQUENCY TRANSLATION

As mentioned in Section II and in Reference 1 the process of frequency translation can be defined mathematically by

$$S(t) = \mathrm{Re}\left[w(t)e^{j2\pi f_c t}\right] \tag{1}$$

where S(t) is the real I. F. signal, Re denotes "the real part of", w(t) is the modulated low pass signal (generally complex), and f_c is the I. F. carrier frequency. If we write w = u+jv and utilize a sampling rate, $r_3 = 4f_c$, then

$$S_n = \mathrm{Re}\left[w_n e^{j\pi n/2}\right] = \mathrm{Re}\left[j^n w_n\right] \tag{2}$$

so that the transmitted sequence, S_o, S_1, S_2, ..., becomes

$$u_o,\ -v_1,\ -u_2,\ v_3,\ u_4,\ -v_5,\ -u_6,\ v_7,\ u_8,\ -v_9, \ldots \quad .$$

By sampling u_o, u_{4k}, u_{8k}, u_{16k}, ... in the real receiver channel and v_1, v_{4k+1}, v_{8k+1}, v_{16k+1}, in the imaginary receiver channel, we obtain samples of $u_{4ik}+jv_{4ik+1}$, which is an approximation to w_{4ik} as discussed in Section IV. By properly combining a number of such samples, better approximations can be attained.

For double sideband amplitude modulation,

$$w(t) = A+m(t) \qquad \text{DSB-AM}$$

where m(t) is the real modulating waveform and A=0 in the suppressed carrier case. Thus

$$S(t) = \left[A+m(t)\right]\cos 2\pi f_c t \qquad \text{DSB-AM}$$

For single sideband amplitude modulation (upper sideband),

$$w(t) = A+m(t)+j\hat{m}(t) \qquad \text{SSB-AM (USB)}$$

where, for the case of true SSB (rather than vestigial sideband), $\hat{m}(t)$ is the Hilbert transform of $m(t)$. The analytic signal $f_+(t) = f(t)+j\hat{f}(t)$, corresponding to any real signal, $f(t)$, is more easily visualized in the frequency domain as

$$F_+(f) = \begin{cases} 2F(f), & f>0 \\ 0, & f<0 \end{cases} \quad . \tag{3}$$

For angle modulation

$$w(t) = Ae^{j\theta(t)} \qquad \text{PM and FM}$$

so that

$$S(t) = A\cos\left[2\pi f_c t+\theta(t)\right] \qquad \text{PM and FM}$$

For phase modulation:

$$\theta(t) = \beta m(t) \qquad \text{PM}$$

where β is called the modulation index, and for frequency modulation

$$\theta(t) = \beta \int^t m(t)dt \qquad \text{FM.}$$

Single sideband phase and frequency modulation can be obtained by setting

$$w(t) = Ae^{j\theta+(t)} \qquad \text{SSB-PM and FM}$$

where $\theta+(t) = \theta(t)+j\hat{\theta}(t)$, so that

$$S(t) = Ae^{-\hat{\theta}(t)}\cos\left[\omega_c t+\theta(t)\right] \qquad \text{SSB-PM and FM.}$$

It can be shown that the spectrum of w(t) is zero for negative frequencies in this case, and hence the spectrum of S(t) is zero for frequencies below the carrier. The modulating signal can be recovered from the receiver's estimate of w(t y

$$A+m(t) = u(t) = \text{Re}\left[w(t)\right] \tag{4}$$

for amplitude modulation (DSB and SSB) and by

$$\theta(t) = \tan^{-1} \frac{v(t)}{u(t)} \tag{5}$$

for angle modulation. Equation (5) applies to single sideband, as well as to conventional double sideband, phase and frequency modulation.

If the modulating signal, m(t), is m-ary (digital data) AM becomes amplitude shift keying (ASK), FM becomes frequency shift keying (FSK), and PM becomes phase shift keying (PSK). For optimum performance, the (sin x)/x spectra of these digital signals would be digitally filtered to produce a spectrum better matched to the channel (e.g., a raised cosine characteristic).

2. SINGLE SIDEBAND AND THE HILBERT TRANSFORM

A ideal double-sideband low-pass filter

$$H_2(f) = \begin{cases} 1, & -B<f<B \\ 0, & \text{elsewhere} \end{cases} \tag{6}$$

has a real impulse response:

$$h_2(t) = 2B \frac{\sin 2\pi Bt}{2\pi Bt} \tag{7}$$

An ideal single-sideband low-pass filter

$$H_1(f) = \begin{cases} 2, & 0<f<B \\ 0, & \text{elsewhere} \end{cases} \tag{8}$$

can be thought of as a B/2 hertz real low-pass filter translated to the right by B/2 hertz. Its impulse response is therefore

$$h_1(t) = 2\left(B\frac{\sin \pi Bt}{\pi Bt}\right)e^{j\pi Bt}. \tag{9}$$

Expanding the complex exponential as cos+jsin, and carrying out the multiplication, we obtain

$$h_1(t) = 2B\left(\frac{\sin 2\pi Bt}{2\pi Bt} + j\,\frac{1-\cos 2\pi Bt}{2\pi Bt}\right) = h_2(t) + j\hat{h}_2(t). \tag{10}$$

As $B \rightarrow \infty$, $h_2(t) \rightarrow \delta(t)$ and $\hat{h}_2(t) \rightarrow 1/(\pi t)$. Thus, in the case of an infinite bandwidth signal, the real channel passes the signal without filtering and the imaginary channel performs a Hilbert transform (since convolution with $1/(\pi t)$ is, by definition, the Hilbert transform). However, since we are dealing with bandlimited signals, the Hilbert transform is not required. What is required is two digital filters designed as a B/2 hertz low-pass filter multiplied by $\cos(\pi Bn/r_1)$ and by $\sin(\pi Bn/r_1)$ respectively, as indicated by equation (9). A single sideband signal must have a spectrum confined to $f_c<f<f_c+B$ for the upper sideband case. Use of the Hilbert transform instead of equation (9) would filter out only the frequencies below the carrier; thus doing only half the job.

If the desired baseband signal is confirmed to the band $b< f <B$ (e.g., 300 to 3,000 hertz), the corresponding complex single sideband filter is

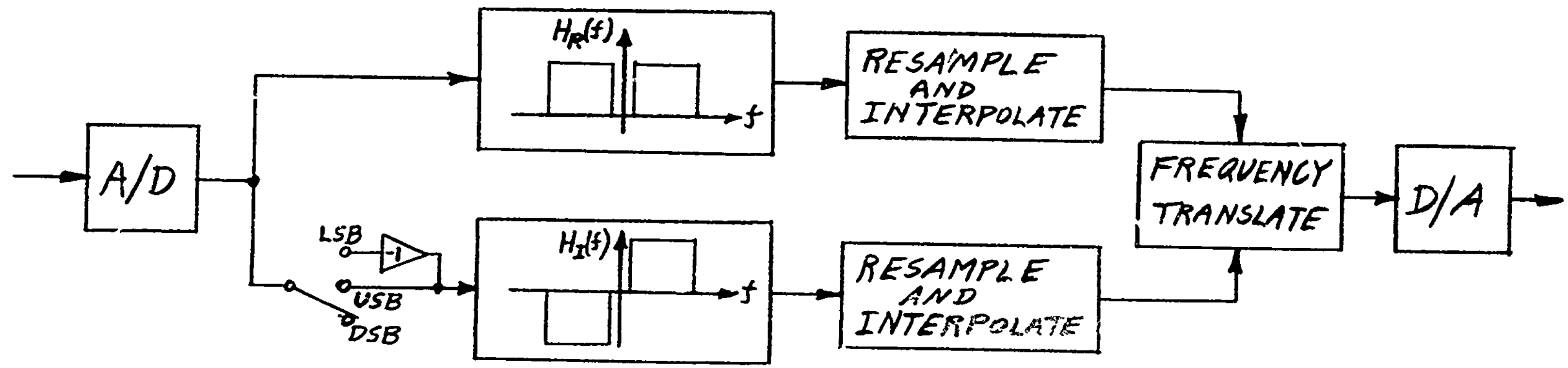
$$h_n = \frac{2B_1}{r_1}\,\frac{\sin(\pi B_1 n/r_1)}{\pi B_1 n/r_1}\cos(\pi B_2 n/r_1) + j\frac{2B_1}{r_1}\,\frac{\sin(\pi B_1 n/r_1)}{\pi B_1 n/r_1}\sin(\pi B_2 n/r_1) \tag{11}$$

where $B_1 = (B-b)/2 = 1,350$ hertz, $B_2 = (B+b)/2 = 1,650$ hertz. Both the real and the imaginary parts of this filter must be further multiplied by an appropriate window function as indicated in Sections III.1 and III.6. The real part of this impulse response can be used for double sideband; it will serve to filter out the carrier in receiver operation. For lower sideband operation it is only necessary to use the negative of the imaginary channel.

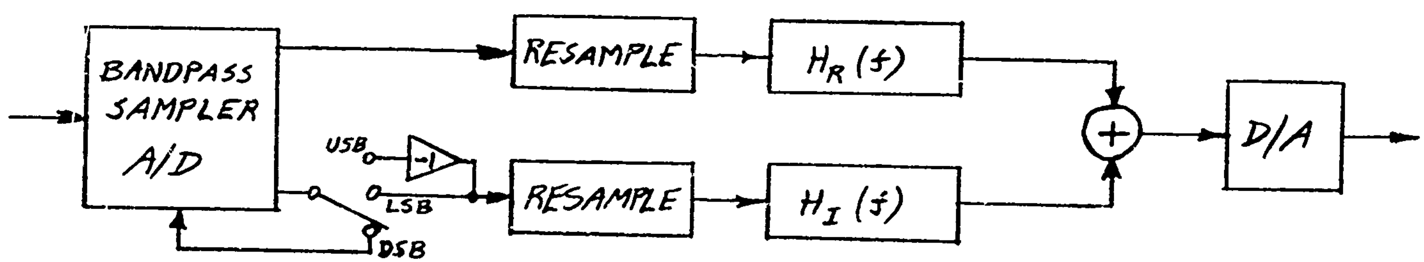
This scheme is illustrated in figure 19. In the transmitter the modulating signal is convolved with $h(t) = h_R(t)+jh_I(t)$ to produce w(t). If $r(t) = r_R(t)+jr_I(t)$ represents the low-pass complex output of the bandpass sampler, the modulating signal is recovered as

$$\mathrm{Re}\left[r(t)*h(t)\right] = r_R(t)*h_R(t) - r_I(t)*h_I(t), \tag{12}$$

where * denotes convolution.



a. Transmitter



b. Receiver

Figure 19: Basic AM Transceiver Configuration

The alternative method of using a single filter response for single sideband and multiplying the signal by cosine and sine, as illustrated in figure 8 of Section I. 2, is equivalent if and only if the delay through the actual filter corresponds to an interger number of cycles of the multiplying sinousoids. Otherwise the phase would not be strictly linear, suffering a discontinuity of $2\pi B_2 M/r_1$ at zero frequency. If the method of Section I. 2 is used M would be a multiple of 5 and B_2 would be 1.6KHz if r_1 = 8KHz. (Or B_2 = 1.65 and r_1 = 8.25).

3. ZERO-QUADRATURE SAMPLING FOR DOUBLE SIDEBAND

The double-sideband receiver configurations in figure 1b and 19b show a phase adjustment feedback loop which attempts to drive the imaginary (or quadrature) samples to zero. For coherent reception (carrier phase know at the receiver) there is no need for such a loop. For non-coherent reception alternative schemes would be (1) to process both the real and the imaginary channels through the double sideband filter (no B/2 frequency shift) and use a read only memory to take the square root of the sum of the squares and (2) to employ stationary point sampling (20, 21).

Suppose the received signal (real channel) is

$$z(t) = A\sin(2\pi ft+\theta)+n(t) \tag{13}$$

where A is to be estimated and n(t) is a stationary, zero-mean, Gaussian process with variance σ^2. If the frequency and phase (f and θ) were known exactly, coherent sampling would produce samples of

$$a = A+n(t) \tag{14}$$

with probability density

$$p(a) = \phi(a-A) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(a-A)^2}{2\sigma^2}} \tag{15}$$

If f is known, but not θ, a quadrature zeroing phase adjustment loop can come very close to attaining the same result.

If z(t) were sampled with random phase the probability density becomes

$$p(z) = \int_{-A}^{A} \frac{\phi(z-x)}{\pi\sqrt{A^2-x^2}}dx = \frac{1}{\pi}\int_{0}^{\pi} \phi(z-A\cos\theta)d\theta. \qquad (16)$$

The measurement is random even in the complete absence of noise ($\sigma = 0$). The square root of the sum of the squares procedure is equivalent to envelope detection and produces the modified Rayliegh (or Rice) density[22]. Stationary point sampling leads to the modified-Normal (or Harley-Abend) density[20, 21]. For large signal to noise ratios ($S/N = A^2/2\sigma^2$) both of these approaches the choerent result of equation (15).

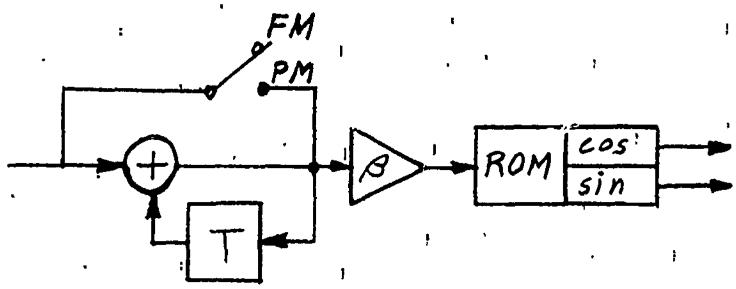
4. ANGLE MODULATION AND PREEMPHASIS

For angle modulation the switch in figure 19a would be in the DSB position and the modulator of figure 20 would be inserted between the real channel interpolator and the frequency translator. The modulation index, β, is set equal to unity for narrowband FM. This produces a complex signal with a non-symmetrical spectrum occupying a 14 KHz band from -7 KHz to +7KHz. The modulator operates at the r_3 sampling rate.

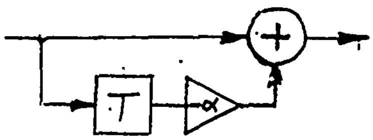
In the computer simulations angle modulation was performed at the r_2 sampling rate and the interpolation to r_3 was performed after the cosine-sine read-only-memory. This had two drawbacks. At the lower sampling rate the z-plane integrator was a poor approximation to true integration and the interpolation filters were required to operate on 2B = 14 KHz signals. This was corrected for the breadboard as shown in figure 11a of Section II. 3.

F. M. preemphasis and deemphasis circuits are shown in figures 20b and c. To prevent distortion of the demodulated signal, preemphasis must be performed before angle modulation, deemphasis must be performed after angle demodulation, and both must be performed at the same sampling rate. A value of 7/8 for the preemphasis parameter would closely approximate current analog receiver practice.

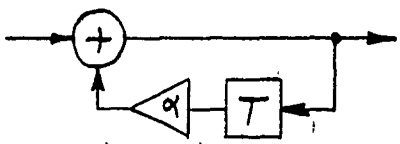
F. M. preemphasis will be performed at the r_2 sampling rate, between the resampling filter and the interpolator. If performed at the low sampling rate, r_1, its frequency response would not be sufficiently linear at the high frequencies. If performed at the high sampling rate, r_3, we would have difficulty in undoing its effect at the receiver. With it at r_2, the deemphasis circuit (also at r_2) will exactly cancel its effect on the demodulated signal, assuming that demodulation is performed before deemphasis. This last requirement is no problem because angle demodulation must be



a. Phase and Frequency Modulator



b. FM Preemphasis



c. FM Deemphasis

Figure 20: Angle Modulation and Preemphasis

performed at r_2. An r_1 = 8KHz sampling rate could not handle a 14KHz signal. Furthermore, the differentiation required for F. M. demodulation is more easily performed at a higher rate (in fact, it can be performed by figure 20b with $\alpha=1$ if the sampling rate were high enough).

5. PHASE AND FREQUENCY DEMODULATION

If the received low-pass complex signal is

$$w(t) = u(t)+jv(t) = A(t)e^{j\theta(t)} \tag{17}$$

then to demodulate phase modulation we must recover

$$\theta(t) = \operatorname{Im}\left[\ln w(t)\right] = \tan^{-1}\frac{v(t)}{u(t)} \qquad \text{(PM)}.$$

To demodulate frequency modulation we must recover

$$\frac{d\theta(t)}{dt} = \operatorname{Im}\left[\frac{\dot{w}(t)}{w(t)}\right] = \frac{u(t)\dot{v}(t)-v(t)\dot{u}(t)}{u^2(t)+v^2(t)} \qquad \text{(FM)}.$$

One approach to angle demodulation is to frequency demodulate as shown in figure 21 and digitally integrate the result when operating with phase modulation. A second alternative is to perform arc tangent phase demodulation as shown in figure 11b of Section II. 3, and to differentiate the result when operating with frequency modulation. With this second alternative deemphasis can be performed at the r_2 sampling rate at the input to the differentiator.

The first alternative was used in the computer simulation and the second alternative was used in the breadboard. Figure 21 does not show the resampling. In the simulations, resampling filters preceeded the differentiators and all computations in figure 21 were performed at the low r_1 sampling rate. This forced us to use a higher basis sampling rate (r_1 = 15KHz) in the simulations which increased the complexity of all filters and degraded the performance of FM. Since differentiation and integration are more easily approximated at a high sampling rate, and since preemphasis in the transmitter was performed at r_2, a better procedure would have been to operate at r_2 and resample after deemphasis.

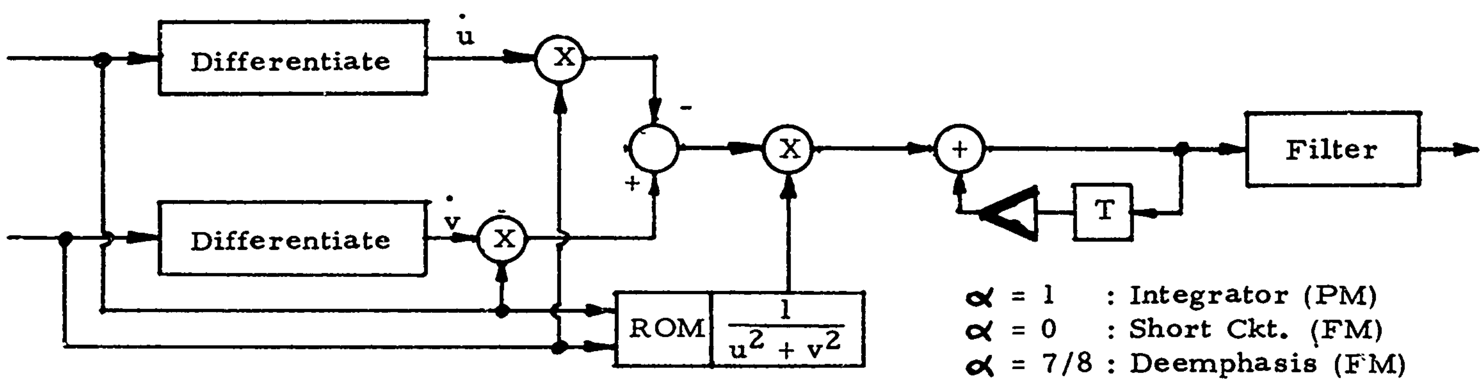


Figure 21: Deriative-Measurement Technique for Angle Demodulation

In the FM mode, figure 11b utilizes a non-recursive low-pass differentiator, approximating

$$H(f) = \begin{cases} j2\pi f, & |f|<B \\ 0\ , & |f|>B, \end{cases}$$

to filter out any energy that might interfere with the r_1 spectral repeats. An alternative (utilized in the breadboard) is to use a simple z-plane differentiator followed by our standard resampling filter.

SECTION VI

TRANSCEIVER DESIGN AND IMPLEMENTATION

This section contains various details revelent to the overall system design (Section II), the computer simulations (Section VII), and the transceiver breadboard (Section VIII), that were not covered elsewhere.

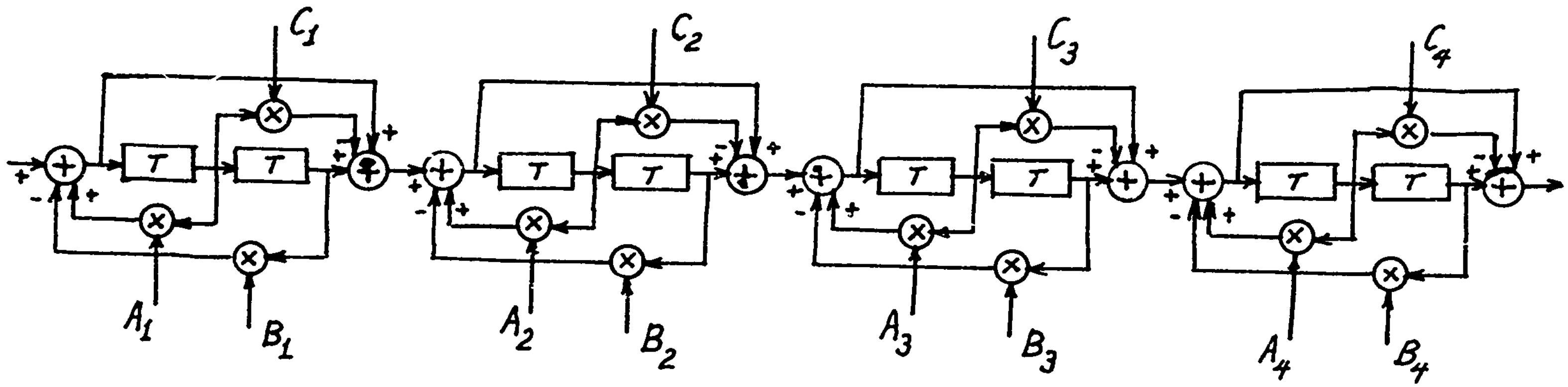
1. RECURSIVE FILTERING

The applicability of recursive filters was investigated in the form of an eighth order (eight poles and eight zeros) Cauer Parameter type C filter. The poles and zeros of this filter were chosen to produce approximately 60 db (57.82db) attenuation in the stopband and 10 percent ripple in the passband with a 20 percent transaction band. The pole-zero locations were obtained from standard filter design handbooks[23] and their actual values in the s-plane are given in figure 22. In transforming from the s-plane to the z-plane the bilinear transformation was used where

$$z = \frac{1+s}{1-s} .$$

and where the values of s were normalized by multiplying by $\tan \pi f_c$ where $f_c = 0.1$ i.e., a cut-off frequency of 1/10th the sampling rate was used. A digital implementation of the filter shown in figure 22 was simulated on the computer. It was found that the filter constants (A_i, B_i, C_i) could be reduced to 8-bit accuracy without significantly altering the filter characteristics. In the passband the ripple was less than 0.5db and in the stopband the attenuation was in excess of 42db. All shift register contents were maintained to 20-bit accuracy and all shift register contents were rounded-off to 12-bits before entering the multipliers. The frequency characteristics of the recursive filter were obtained by performing a Fast Fourier Transform (FFT) of the impulse response. Figure 23 is a plot of the frequency response of this filter and figure 24 is a brief computer printout of the magnitude and phase response (the first and second data columns). The third and fourth columns show the magnitude in db and the normalized frequency (normalized to unity at the cutoff frequency).

From the example presented it is clearly seen that extremely stringent filter characteristics can be realized with digital recursive filters employing a relatively small number of shift registers (storage) and arithmetic operations (especially multiplications) per unit sampling time. The main



S-PLANE POLES (S_p) & ZEROS (S_o)

$S_{p1,2} = -0.0396068350 \pm j\,1.0306888432$

$S_{p3,4} = -0.1475747947 \pm j\,0.9550996330$

$S_{p5,6} = -0.3321414712 \pm j\,0.7360558243$

$S_{p7,8} = -0.5498131652 \pm j\,0.2837282021$

$S_{o1,2} = \pm j\,1.2309850902$

$S_{o3,4} = \pm j\,1.3723325582$

$S_{o5,6} = \pm j\,1.9314481827$

$S_{o7,8} = \pm j\,\infty$

QUANTIZED RECURSIVE FILTER CONSTANTS (x64)

$A_1 = 100$	$B_1 = 61$	$C_1 = 93$
$A_2 = 97$	$B_2 = 54$	$C_2 = 86$
$A_3 = 93$	$B_3 = 42$	$C_3 = 56$
$A_4 = 87$	$B_4 = 31$	$C_4 = -128$

Figure 22: Caver Parameter Recursive Filter

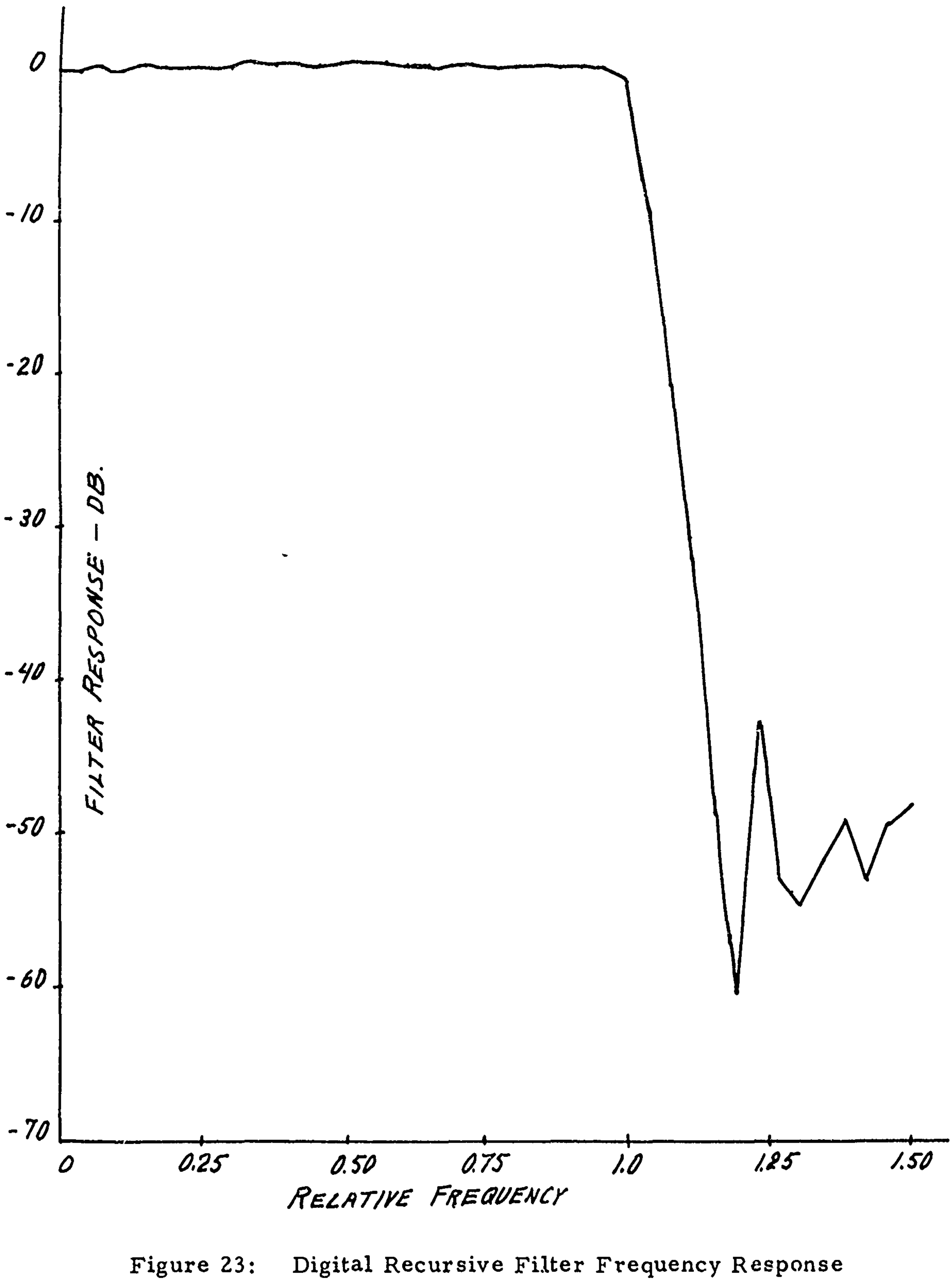


Figure 23: Digital Recursive Filter Frequency Response

	MAGNITUDE	PHASE	MAG.(DB)	REL.FREQ.
1	31.0978	0.000	0.000	0.000
2	30.5559	-7.998	-0.153	0.039
3	31.2301	-17.515	0.037	0.078
4	30.9331	-27.790	-0.046	0.117
5	31.4030	-36.861	0.085	0.156
6	31.3263	-46.008	0.064	0.195
7	31.1741	-54.850	0.021	0.234
8	31.3141	-64.839	0.060	0.273
9	31.5860	-74.952	0.135	0.313
10	32.6866	-85.283	0.433	0.352
11	31.7208	-96.638	0.172	0.391
12	31.9420	-106.024	0.233	0.430
13	31.9560	-118.300	0.236	0.469
14	32.3525	-129.491	0.344	0.508
15	32.4445	-141.664	0.368	0.547
16	32.5667	-154.171	0.401	0.586
17	32.1814	-168.468	0.298	0.625
18	32.2536	177.597	0.317	0.664
19	31.9402	163.219	0.232	0.703
20	32.1787	146.765	0.297	0.742
21	31.5674	129.395	0.130	0.781
22	31.6212	110.289	0.145	0.820
23	31.7581	89.268	0.182	0.859
24	31.2529	63.873	0.043	0.898
25	31.4117	33.713	0.087	0.938
26	31.2508	-6.890	0.043	0.977
27	28.5363	-68.259	-0.747	1.016
28	10.9751	-133.115	-9.046	1.055
29	2.8668	-172.871	-20.707	1.094
30	0.7253	146.830	-32.644	1.133
31	0.1121	176.226	-48.860	1.172
32	0.0298	5.036	-60.381	1.211
33	0.2289	-13.497	-42.661	1.250
34	0.0697	17.711	-52.985	1.289
35	0.0586	-28.111	-54.490	1.328
36	0.0793	-59.611	-51.872	1.367
37	0.1102	91.122	-49.008	1.406
38	0.0694	107.007	-53.034	1.445
39	0.1062	70.319	-49.334	1.484
40	0.1225	53.984	-48.092	1.523

END RCRFLT 15.5 SEC.

Figure 24: Recursive Filter Computer Printout

disadvantages of recursive filters, however, are (a) somewhat higher bit accuracy is required in the arithmetic operations as compared to non-recursive filters, (b) significantly higher bit accuracy is required in the storage shift registers, (c) the phase characteristic is extremely non-linear near the cutoff region unlike the zero differential phase delay characteristics of non-recursive filters, and (d) the sequence of arithmetic operations to be performed is such that implementation of recursive digital filters does not readily lend itself to the pipeline processing technique. Because of these disadvantages it was concluded that recursive digital filters would not be appropriate for the multimode transceiver design.

Rabiner, et, al [11, 12, 24] have presented an interesting recursive implementation for finite response filters that have been designed by the frequency sample specification approach. These filters require more storage than, and approximately as many multiplications as, the convolutional (non-recursive) implementation. They have all the disadvantages of standard recursive filters except (c).

To clarify the accuracy problem, consider a simple recursive filter such as in figure 20c of Section V.4. If the input and the stored constant, α, are each 10 bits, the result of multiplying the two requires 20 bits of storage. If no truncation or rounding is used, a 30 bit number is required for the next time around the loop. And this would continue indefinitely. If we assume we use a 10 bit by 10 bit multiplier, we must round to ten bits each time around the loop. A convolutional filter (such as shown in figure 2 of Section II.1), however, need never round as long as the accumulator can hold a 20 bit number.

Recursive implementation was used for the interpolation filters because they could be realized without multiplications.

2. NON-RECURSIVE FILTERING

The interactive computer program discussed in Section III.6 was used to compute the quantized values of the tap weights for all convolutional filters used in the simulations and the breadboard. CONFIDE is a time-shared Fortran program for the design and frequency analysis of finite response filters with zero differential delay (absolutely linear phase). A simplified flow chart is given in figure 25 and the program itself is given in figure 26. The operator first enters C, L, and A, where C is the window parameter, L is the number of bits precision for the tap weights, and $A = \alpha$ is the raised cosine roll-off parameter (A=0 for no roll-off). The tap weights are normalized to a maximum value of $c_0 = 2^{L-1} - 1$ and rounded to the nearest integer. Thus, L=10 represents tap weights ranging from -511 to +511 (or $\pm\frac{511}{512}$ in steps of $E = \frac{1}{512}$). While the Hamming window (C = .54)

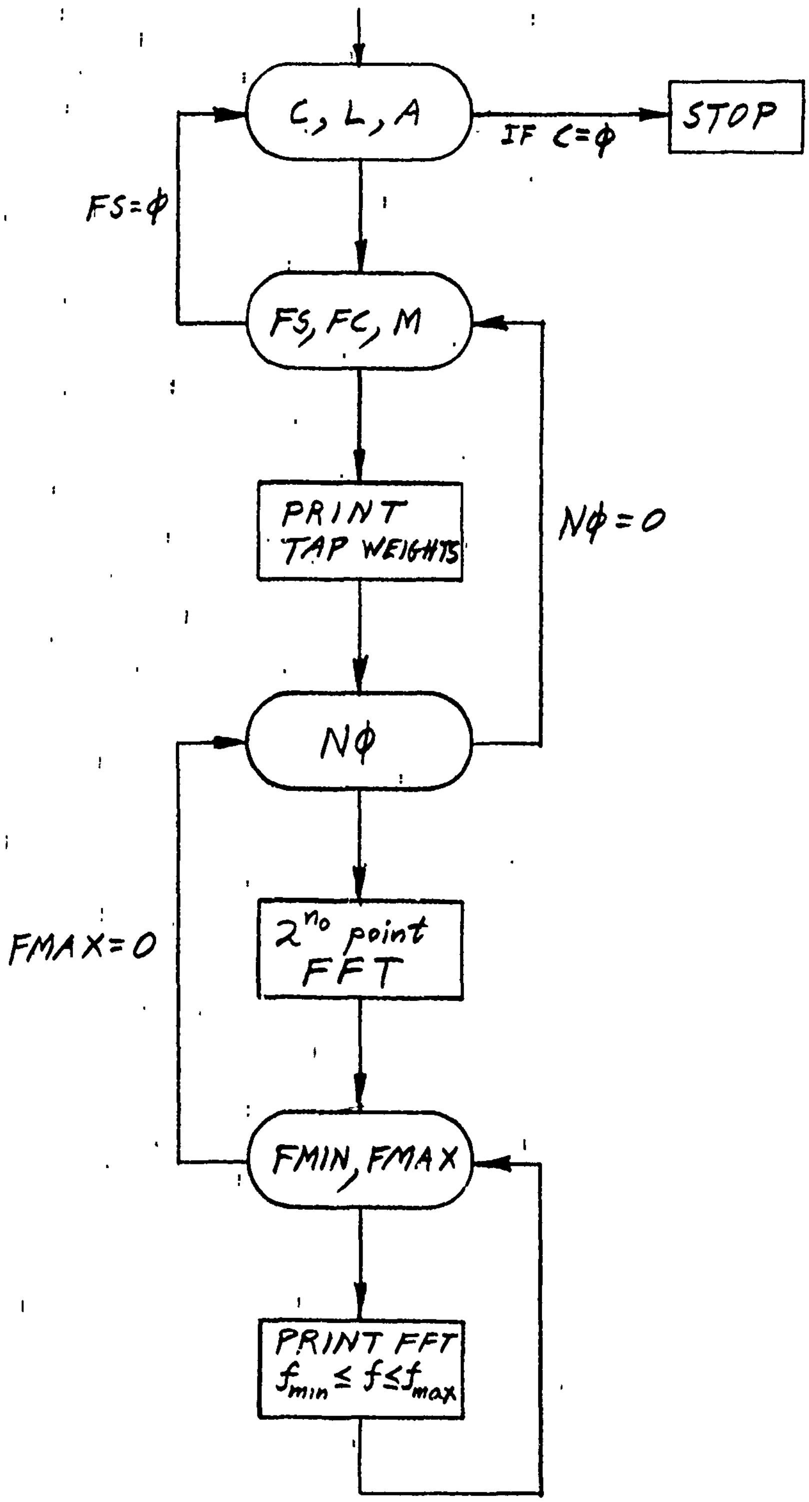


Figure 25: Simplified Flowchart of Convolutional Filter Design Program (CONFIDE) (Oval blocks represent Teletype inputs)

```
C-  (2**N0)PT. FFT OF A (2M+1)STAGE L BIT CONVOLUTIONAL FILTER
C-
C-      WINDOWS-
C-        HAMMING: C=.54
C-        HANNING: C=.5
C-        G&R HAM: C=.56
C-      A = RAISED COSINE ROLL-OFF
C-      FC=0 FOR DIFFERENTIATION
C-
 DIMENSION G(512)
  INTEGER H(512),S,SUM
COMPLEX X(2,2048),W
PI=3.14159265
 75 PRINT 15
 15 FORMAT ("C,L,A")
 READ /,C,L,A
 IF (C=0) GO TO 44
 S=2**(L-1)-1
  PRINT 12 (S)
  12 FORMAT ("     MAXIMUM TAP =",I7)
 60 PRINT 16
 16 FORMAT ("FS,FC,M")
 READ /,FS,FC,M
 IF (FS=0) GO TO 75
 G0=2*FC/FS
 IF (FC=0) G0=FS/(2*PI)
 R=S/G0
 SUM = S
 P=G0*G0
 IF(FC=0) P=0
DO 30 I=1,M
 PII = PI*I
 Q=C+(1.-C)*COS(PII/M)
 IF(FC=0) GO TO 28
 E = A*G0*I
 IF (E.NE.0.5) V = COS(PI*E)/(1. -4*E**2)
 IF (E=0.5) V=PI/4.
 Q = Q*V
 G(I)=Q*SIN(G0*PII)/(PII)
 GO TO 29
 28 G0=-G0
 G(I)=Q*G0/I
 29 CONTINUE
 IF (G(I).GE.0)H(I)=R*G(I)+.5
 IF (G(I).LT.0) H(I)=R*G(I)-.5
 G(I)=H(I)/R
 SUM = SUM + 2*H(I)
 P=P+2.*G(I)*G(I)
```

Figure 26: Convolutional Filter Design (CONFIDE) Fortran Program

```
 30 CONTINUE
 PRINT 70 (H(I),I=1,M)
 SSQ=P/(G0*G0)
 SRS=SQRT(SSQ)
 IF(FC=0) SUM=0
 PRINT 36 (SUM,SSQ,SRS)
 36 FORMAT ("SUM=",I7,"  SSQ=" ,F8.3,"  SRS=",F8.3)
 85 PRINT 84
 84 FORMAT ("N0")
 READ /,N0
 IF (N0=0) GO TO 60
 N=2**N0
 M1=2*M+1
 PRINT 40 (N,M1,L)
 AA=1
 IF(FC=0) AA=-1
 X(1,1)=CMPLX(G0,0.)
 IF (FC=0) X(1,1)=CMPLX(0.,0.)
 DO 17 I=1,M
 X(1,I+1)=CMPLX(G(I),0.)
 17 X(1,N-I+1)=AA*CMPLX(G(I),0.)
 NM=N-M-2
DO 2 I=M,NM
 2 X(1,I+2)=(0.,0.)
 N2=N/2
 P2=2.*PI/N
DO 5 J=1,N0
N2J=N/(2**J)
NK=N2J
NI=(2**J)/2
DO 4 I=1,NI
IN2J =  (I-1)*N2J
T=IN2J*P2*(-1.)
W=CMPLX(COS(T),SIN(T))
DO 4 K=1,NK
ISUB = K+IN2J
ISUB1 = K+IN2J*2
ISUB2 = ISUB1+N2J
ISUB3 = ISUB + N2
X(2,ISUB)=X(1,ISUB1)+W*X(1,ISUB2)
X(2,ISUB3)=X(1,ISUB1)-W*X(1,ISUB2)
4CONTINUE
DO 5 K=1,N
5X(1,K)=X(2,K)
 60 PRINT 35
 35 FORMAT ("FMIN,FMAX")
 READ /,FMIN,FMAX
 IF (FMAX=0) GO TO 85
```

Figure 26 (Continued)

```
7200  KMIN =N*FMIN/FS+1
7300  KMAX =N*FMAX/FS+2
7350  SS=0
7400 DO 6 K = KMIN,KMAX
7500  Y=CABS(X(1,K))
7600  Z=REAL(X(1,K))
7700  D = 20.*ALOG10(Y)
7800  F=FS*(K-1)/N
7850  IF (FC=0) GO TO 92
7900   PRINT 20 (K,Z,D,F)
7920  GO TO 6
7940  92 U=AIMAG(X(1,K))
7945  B=0
7950  IF(F.GT.0) B=(U-F)/F
7960  SS=SS+(U-F)*(U-F)
7970  AMS=SS/(K-KMIN+1)
7980  PRINT 22 (K,F,U,B,AMS)
8000  6 CONTINUE
8100  GO TO 80
8200  40 FORMAT (I4," POINT TRANSFORM OF ",I3," STAGE ",I3," BIT\\
         \\ FILTER"/)
8300  20 FORMAT (I3,F12.4,F13.3,F10.3)
8350  22 FORMAT (I3,3F12.3,F15.6)
8400  70 FORMAT (10I7)
8500  44 STOP
8600  END
```

Figure 26 (Concluded)

is superior for high precision applications, for $L \leq 10$ the Hanning window ($C = .5$) was found to aid in the natural truncation phenomenon discussed in Section III. 6. While a filter with 8 bit precision was able to meet the specs of at least 40db attenuation in the stopband with a transition band of only 300 hertz, 10 bit precision was chosen for the tap weights to provide a safety margin for the additional transceiver processing that must be performed.

The next inputs are FS, FC, and M, where FS=r is the sampling rate and FC is the 6db cut-off frequency for the version of the filter symmtrical about zero frequency, (i. e., FC $\gtrsim$ 1.5KHz for 3KHz single sideband). If FC=0, the tap weights for a differentiating filter are calculated. The M filter tap weights (C_1, $C_2, \ldots, C_M$ in Section III) are computed and printed ten per line. The number of nonzero taps will generally be less than 2M+1. The next input, $N_o = n_o$, determines that a 2^{no} point Fast Fourier Transform of the filter's unit response will be taken. The inputs FMIN and FMAX specify the range of frequencies to be printed out. From figure 25 we see that the operator can examine any portion of the frequency response in any detail desired (up to a 2048 point transform) before modifying the filter.

The computer simulations (Section VII) employed a basic sampling rate of r_1=15KHz because at the time they were performed we believed it would be more efficient to do frequency modulation at the low rate. The resampled rate for the simulations, r_2=120KHz, was also unnecesarily high. The earlier version of CONFIDE used to design the simulation filters did not have provision for the raised cosine roll-off. These deficiencies combined to make the filters have at least twice the number of taps that they would have required at a more reasonable sampling rate. Sample printouts are given in figures 27 through 30. The specs on the SSB filter required a drop from -3db to -40db in 300 hertz. The frequency response in figure 31 shows that these specifications are met. The differentiating filter in figure 30 is highly inefficient in that it fails to take advantage of a recent "breakthrough" discussed in Section VI. 3.

Since ripple tends to accumulate from the cascade of all transmitter and receiver filters, the specs on the SSB filter were revised for the breadboard to call for a drop from 1db to 50db in 450 hertz. The corresponding frequency response and CONFIDE printouts, for a 8KHz sampling rate, is given in figures 32 and 33. A two-to-one resampling filter used for the breadboard demonstration is shown in figure 34.

To produce a double sideband filter characteristic with a passband from 300 to 3,000 hertz as mentioned in Section V. 2, the CONFIDE program of figure 26 was modified by adding two instructions:

C,L
?.5,10
MAXIMUM TAP = 511
FS,FC,M
?15,1.42,50

481	397	277	146	28	-57	-100	-101	-71	-26
18	47	55	45	22	-4	-24	-33	-29	-17
-2	12	19	-19	13	4	-5	-10	-11	-8
-3	1	5	6	5	2	0	-2	-2	-2
-1	0	0	1	0	0	0	0	0	0

SSQ= 3.027 SRS= 1.740
NO
?8
256 POINT TRANSFORM OF ~~101~~ 89 STAGE 10 BIT FILTER

FMIN,FMAX
?0,3

		Gain (db)	Frequency (KHz)
1	1.0008	0.007	0.000
2	1.0005	0.004	0.059
3	0.9998	-0.002	0.117
4	0.9988	-0.011	0.176
5	0.9978	-0.019	0.234
6	0.9974	-0.022	0.293
7	0.9981	-0.017	0.352
8	0.9994	-0.005	0.410
9	1.0008	0.007	0.469
10	1.0013	0.012	0.527
11	1.0008	0.007	0.586
12	1.0001	0.001	0.645
13	1.0001	0.001	0.703
14	1.0010	0.008	0.762
15	1.0015	0.013	0.820
16	1.0005	0.004	0.879
17	0.9985	-0.013	0.938
18	0.9982	-0.015	0.996
19	1.0021	0.018	1.055
20	1.0067	0.058	1.113
21	0.9996	-0.004	1.172
22	0.9611	-0.345	1.230
23	0.8730	-1.180	1.289
24	0.7300	-2.733	1.348
25	0.5468	-5.243	1.406
26	0.3544	-9.009	1.465
27	0.1882	-14.508	1.523
28	0.0724	-22.807	1.582
29	0.0115	-38.754	1.641
30	-0.0076	-42.356	1.699

Figure 27: Single Sideband Filter, r = 15KHz

```
 31      -0.0060        -44.496      1.758
 32      -0.0005        -65.690      1.816
 33       0.0012        -58.406      1.875
 34      -0.0002        -74.088      1.934
 35      -0.0014        -57.089      1.992
 36      -0.0008        -62.396      2.051
 37       0.0008        -61.476      2.109
 38       0.0018        -54.870      2.168
 39       0.0016        -55.919      2.[illegible]7
 40       0.0009        -60.457      2.285
 41       0.0007        -62.738      2.344
 42       0.0011        -59.421      2.402
 43       0.0014        -56.784      2.461
 44       0.0014        -56.949      2.520
 45       0.0010        -59.694      2.578
 46       0.0006        -64.232      2.637
 47       0.0003        -69.797      2.695
 48       0.0001        -79.347      2.754
 49      -0.0000        -92.508      2.812
 50       0.0002        -75.481      2.871
 51       0.0009        -61.088      2.930
 52       0.0019        -54.608      2.988
 53       0.0024        -52.445      3.047
FMIN,FMAX
?0,0
NO
?10
1.35,1.35
1024 POINT TRANSFORM OF 101 STAGE  10 BIT FILTER


FMIN,FMAX
 93      0.7300          -2.733      1.348
 94      0.6870          -3.261      1.362
FMIN,FMAX
?1.65,1.65
113      0.0115         -38.754      1.641
114      0.0036         -48.812      1.655
FMIN,FMAX
?0,0
NO
?0
FS,FC,M
?0,0,0
C,L
?0,0
```

Figure 27 (Continued)

```
FS,FC,M
?15,3.2,48
     371      84     -97     -75      30      60       2     -43     -19      25
      25     -10     -24      -2      19       9     -12     -12       4      12
       1      -9      -5       5       6      -2      -6      -1       4       2
      -2      -3       1       2       0      -1      -1       1       1       0
       0       0       0       0       0       0       0       0
NO
?0
FS,FC,M
?15,3.2,49
     371      84     -97     -75      30      60       2     -43     -19      26
      25     -10     -24      -2      19       9     -12     -13       4      12
       1      -9      -5       6       6      -2      -6      -1       4       2
      -2      -3       1       2       0      -2      -1       1       1       0
      -1       0       0       0       0       0       0       0       0
NO
?0
FS,FC,M
?15,3.2,48
     371      84
?7
```

128 POINT TRANSFORM OF ~~97~~ 79 STAGE 10 BIT FILTER

```
FMIN,FMAX
?0,3.8[illegible]4
   1        0.9944        -0.048      0.000
   2        0.9951        -0.043      0.117
   3        0.9993        -0.006      0.234
   4        1.0029         0.025      0.352
   5        0.9993        -0.006      0.469
   6        0.9967        -0.029      0.586
   7        1.0012         0.010      0.703
   8        1.0030         0.026      0.820
   9        1.0002         0.002      0.938
  10        1.0009         0.008      1.055
  11        1.0020         0.017      1.172
  12        0.9998        -0.001      1.289
  13        1.0009         0.007      1.406
  14        1.0022         0.019      1.523
  15        0.9985        -0.013      1.641
  16        0.9978        -0.019      1.758
  17        1.0015         0.013      1.875
  18        1.0009         0.008      1.992
  19        0.9999        -0.001      2.109
  20        1.0029         0.025      2.227
```

Figure [illegible]: Double Sideband Filter, r = 15KHz

21	1.0023	0.020	2.344
22	1.0006	0.005	2.461
23	1.0024	0.021	2.578
24	0.9995	-0.005	2.695
25	1.0018	0.015	2.812
26	1.0026	0.023	2.930
27	0.9011	-0.905	3.047
28	0.6131	-4.250	3.164
29	0.2563	-11.826	3.281
30	0.0380	-28.408	3.398
31	-0.0076	-42.358	3.516
32	0.0014	-56.851	3.633
33	-0.0008	-61.567	3.750
34	-0.0012	-58.063	3.867
35	0.0032	-49.914	3.984
36	0.0027	-51.346	4.102

```
FMIN,FMAX
?0,0
NO
?0
```

Figure 28 (Continued)

```
FS,FC,M
?120,7.55,46
    497    457    395    317    231    144     64     -3    -53    -83
    -95    -91    -74    -50    -23      2     23     35     41     39
     32     21      9     -2    -10    -15    -17    -15    -12     -8
     -3      1      3      5      5      4      3      2      1      0
      0      0      0      0      0      0
SSQ=   4.193  SRS=    2.048
NO
?9
 512 POINT TRANSFORM OF  93 STAGE  10 BIT FILTER

7,7
FMIN,FMAX
 30        0.7689        -2.282      6.797
 31        0.6916        -3.204      7.031
FMIN,FMAX
?9.6,15
 41        0.0197       -34.117      9.375
 42        0.0038       -48.481      9.609
 43       -0.0046       -46.720      9.844
 44       -0.0075       -42.460     10.078
 45       -0.0070       -43.117     10.312
 46       -0.0047       -46.609     10.547
 47       -0.0019       -54.398     10.781
 48        0.0005       -66.836     11.016
 49        0.0020       -54.086     11.250
 50        0.0026       -51.798     11.484
 51        0.0024       -52.394     11.719
 52        0.0018       -55.127     11.953
 53        0.0009       -60.538     12.187
 54        0.0002       -73.113     12.422
 55       -0.0002       -72.201     12.656
 56       -0.0004       -67.719     12.891
 57       -0.0003       -69.830     13.125
 58       -0.0001       -81.361     13.359
 59        0.0002       -75.159     13.594
 60        0.0004       -69.107     13.828
 61        0.0004       -68.454     14.062
 62        0.0002       -72.192     14.297
 63       -0.0000      -101.346     14.531
 64       -0.0003       -70.024     14.766
 65       -0.0006       -64.517     15.000
 66       -0.0008       -62.194     15.234
```

use FC=7.6

Figure 29: 8 to 1 Resampling Filter, r = 120 KHz

C,L
?.5,10
MAXIMUM TAP = 511
FS,FC,M
?15,0,51

-511	255	-169	126	-100	82	-70	60	-53	46
-41	37	-33	30	-27	25	-23	21	-19	17
-15	14	-13	12	-11	10	-9	8	-7	6
-6	5	-4	4	-3	3	-2	2	-2	1
-1	1	-1	1	0	0	0	0	0	0
0									

SSQ= 1.570 SRS= 1.253
NO
?7
128 POINT TRANSFORM OF ~~103~~ 89 STAGE 10 BIT FILTER

FMIN,FMAX
?0,7.5

	f	H(f)/j	$\frac{H(f)}{jf} - 1$	Mean Squared Error
1	0.000	0.000	0.000	0.000000
2	0.117	0.110	-0.059	0.000410
3	0.234	0.241	0.029	0.000403
4	0.352	0.361	0.026	0.000499
5	0.469	0.475	0.013	0.000453
6	0.586	0.600	0.024	0.000712
7	0.703	0.720	0.025	0.001023
8	0.820	0.832	0.014	0.001041
9	0.938	0.933	-0.005	0.000931
10	1.055	1.045	-0.009	0.000921
11	1.172	1.179	0.006	0.000870
12	1.289	1.281	-0.006	0.000843
13	1.406	1.370	-0.026	0.001694
14	1.523	1.515	-0.005	0.001607
15	1.641	1.649	0.005	0.001537
16	1.758	1.739	-0.010	0.001629
17	1.875	1.869	-0.003	0.001545
18	1.992	2.013	0.011	0.001676
19	2.109	2.115	0.003	0.001599
20	2.227	2.222	-0.002	0.001525
21	2.344	2.343	-0.000	0.001449
22	2.461	2.460	-0.001	0.001381
23	2.578	2.584	0.002	0.001334
24	2.695	2.701	0.002	0.001287
25	2.812	2.808	-0.002	0.001242
26	2.930	2.927	-0.001	0.001195
27	3.047	3.050	0.001	0.001152
28	3.164	3.170	0.002	0.001119
29	3.281	3.283	0.001	0.001080
30	3.398	3.401	0.001	0.001045

Figure 30: Differentiating Filter, r = 15KHz

31	3.516	3.540	0.007	0.001180
32	3.633	3.651	0.005	0.001232
33	3.750	3.737	-0.003	0.001235
34	3.867	3.873	0.001	0.001206
35	3.984	4.006	0.005	0.001283
36	4.102	4.094	-0.002	0.001259
37	4.219	4.212	-0.002	0.001235
38	4.336	4.335	-0.000	0.001201
39	4.453	4.437	-0.004	0.001231
40	4.570	4.574	0.001	0.001202
41	4.688	4.705	0.004	0.001236
42	4.805	4.808	0.001	0.001207
43	4.922	4.942	0.004	0.001264
44	5.039	5.056	0.003	0.001294
45	5.156	5.140	-0.003	0.001314
46	5.273	5.271	-0.000	0.001285
47	5.391	5.407	0.003	0.001310
48	5.508	5.530	0.004	0.001372
49	5.625	5.663	0.007	0.001597
50	5.742	5.751	0.002	0.001578
51	5.859	5.834	-0.004	0.001654
52	5.977	5.967	-0.002	0.001637
53	6.094	6.089	-0.001	0.001609
54	6.211	6.220	0.001	0.001591
55	6.328	6.360	0.005	0.001722
56	6.445	6.455	0.001	0.001705
57	6.563	6.568	0.001	0.001678
58	6.680	6.701	0.003	0.001717
59	6.797	6.815	0.003	0.001734
60	6.914	6.932	0.003	0.001750
61	7.031	6.988	-0.006	0.001988
62	7.148	7.176	0.004	0.002058
63	7.266	7.221	-0.006	0.002295
64	7.383	5.136	-0.304	0.685850
65	7.500	0.000	-1.000	8.175133
66	7.617	-5.136	-1.674	29.402576

FMIN,FMAX

Figure 30 (Continued)

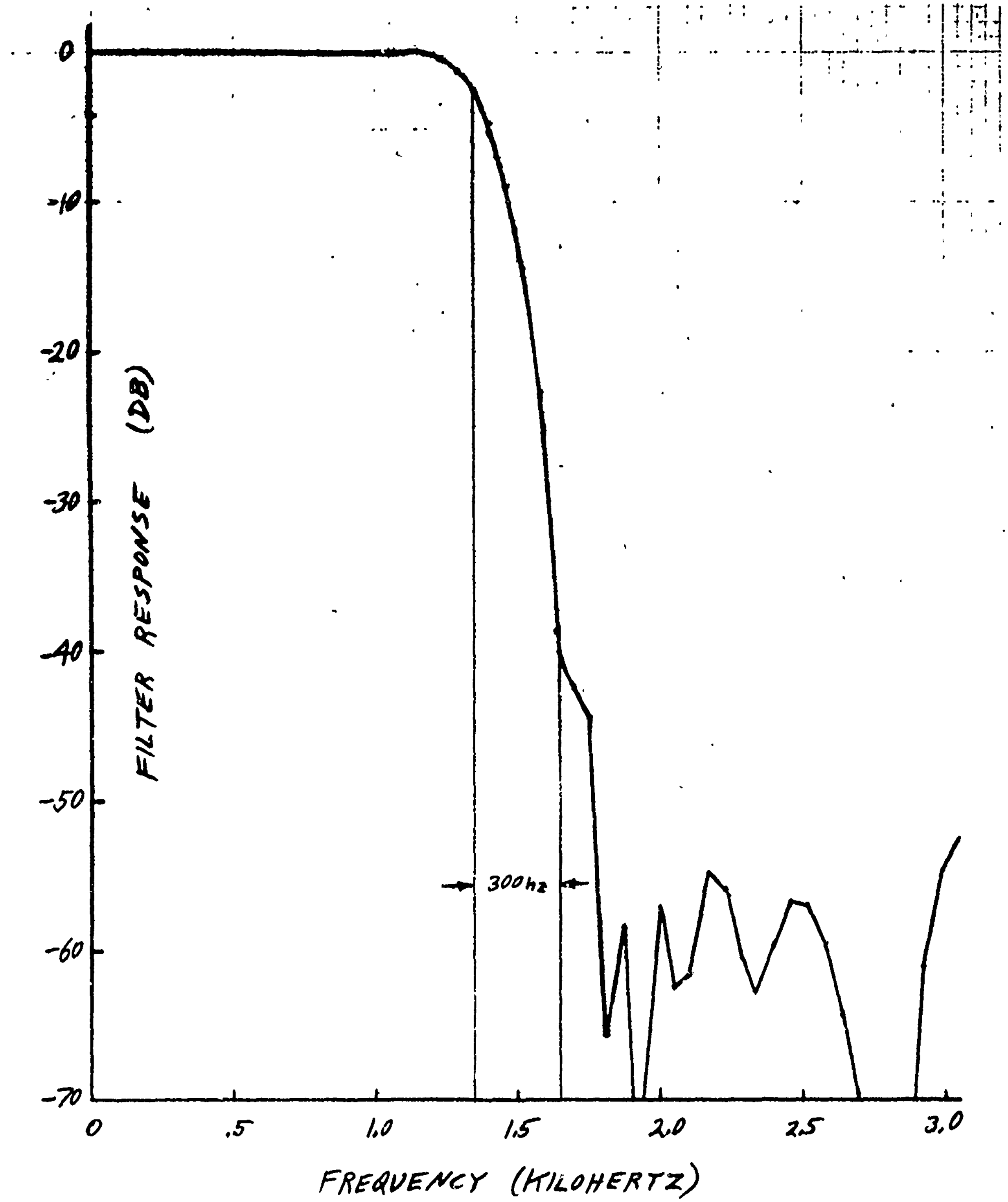


Figure 31: Frequency Response of 89 Stage S.S.B. Non-Recursive Filter, r = 15KHz

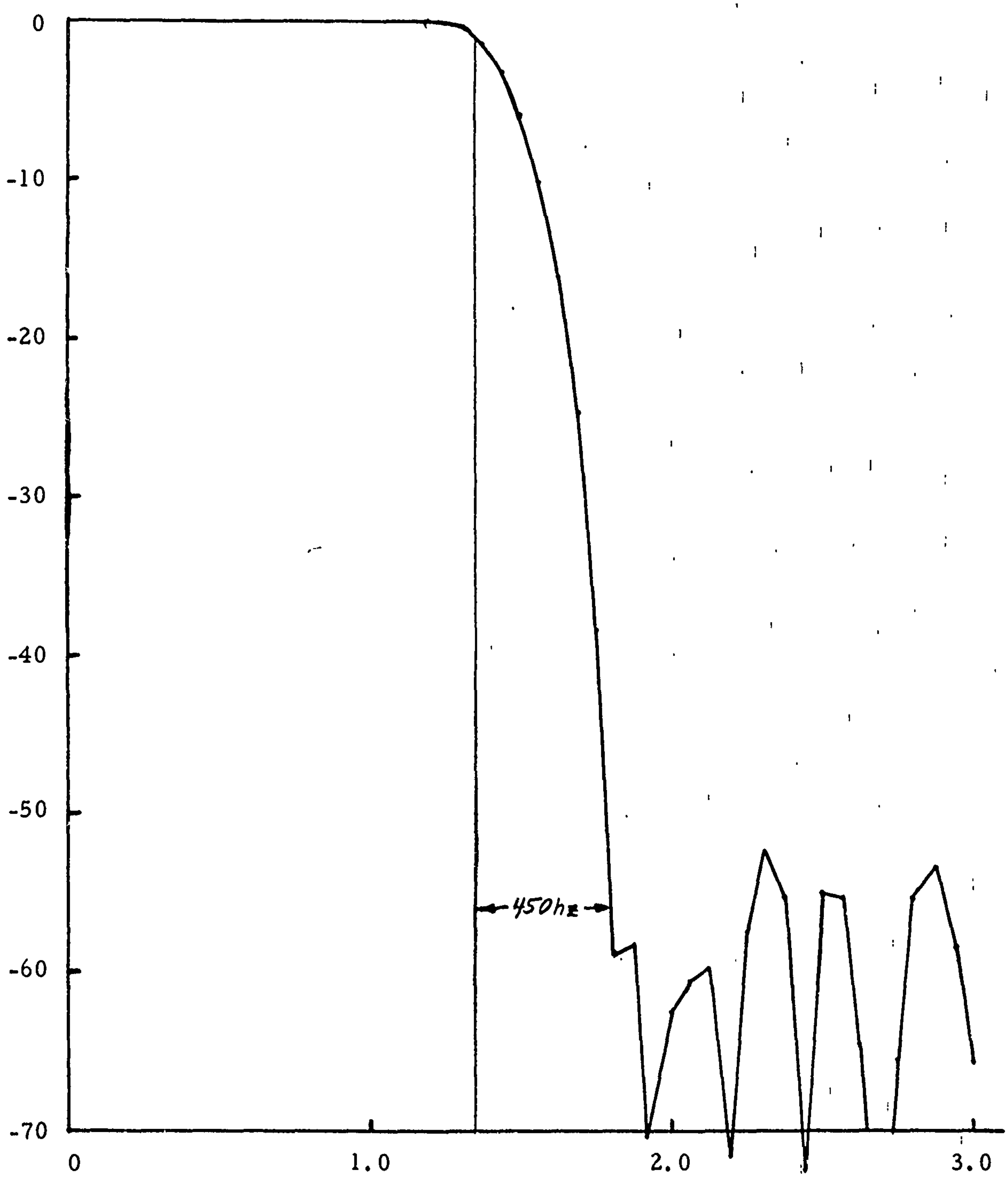


Figure 32: Frequency Response of 47 Stage S.S.B. Non-Recursive Filter, r = 8KHz

```
C,L,A
?.5,10,.1
    MAXIMUM TAP =     511
FS,FC,M
?8,1.5,29
    399    151    -53   -101    -30     44     46      0    -31    -20
      9     19      6     -9     -9      0      6      3     -1     -3
     -1      1      1      0      0      0      0      0      0
SUM=   1365  SSQ=   2.547  SRS=   1.596
NO
?7
 128 POINT TRANSFORM OF  59 STAGE  10 BIT FILTER

FMIN,FMAX                      Gain      Frequency
?0,4                           (db)        (KHz)
  1       1.0017              0.015        0.000
  2       1.0013              0.011        0.063
  3       1.0006              0.005        0.125
  4       1.0003              0.002        0.188
  5       1.0005              0.005        0.250
  6       1.0007              0.006        0.313
  7       1.0003              0.003        0.375
  8       0.9999             -0.001        0.438
  9       1.0002              0.002        0.500
 10       1.0012              0.011        0.563
 11       1.0018              0.015        0.625
 12       1.0008              0.007        0.688
 13       0.9991             -0.008        0.750
 14       0.9983             -0.015        0.813
 15       0.9991             -0.007        0.875
 16       1.0003              0.002        0.938
 17       1.0000             -0.000        1.000
 18       0.9989             -0.010        1.062
 19       0.9991             -0.008        1.125
 20       0.9989             -0.009        1.187
 21       0.9868             -0.115        1.250
 22       0.9419             -0.520        1.312
 23       0.8443             -1.470        1.375
 24       0.6901             -3.222        1.437
```

Figure 33: Single Sideband Filter, γ = 8KHz

25	0.4992	-6.035	1.500
26	0.3087	-10.210	1.562
27	0.1552	-16.181	1.625
28	0.0577	-24.773	1.687
29	0.0119	-38.472	1.750
30	-0.0011	-58.951	1.812
31	-0.0012	-58.315	1.875
32	0.0003	-70.338	1.937
33	0.0007	-62.688	2.000
34	0.0009	-60.793	2.062
35	0.0010	-59.874	2.125
36	0.0003	-71.657	2.187
37	-0.0013	-57.549	2.250
38	-0.0024	-52.419	2.312
39	-0.0017	-55.248	2.375
40	0.0002	-72.661	2.437
41	0.0018	-55.036	2.500
42	0.0017	-55.353	2.562
43	0.0006	-64.761	2.625
44	-0.0001	-80.855	2.687
45	0.0005	-65.747	2.750
46	0.0017	-55.364	2.812
47	0.0021	-53.457	2.875
48	0.0012	-58.575	2.937
49	-0.0005	-66.417	3.000
50	-0.0017	-55.349	3.062
51	-0.0019	-54.278	3.125
52	-0.0014	-57.032	3.187
53	-0.0007	-62.616	3.250
54	-0.0004	-69.043	3.312
55	-0.0004	-68.614	3.375
56	-0.0007	-62.793	3.437
57	-0.0011	-58.919	3.500
58	-0.0011	-59.489	3.562
59	-0.0000	-87.264	3.625
60	0.0017	-55.282	3.687
61	0.0032	-49.974	3.750
62	0.0031	-50.212	3.812
63	0.0012	-58.124	3.875
64	-0.0011	-58.965	3.937
65	-0.0022	-53.145	4.000
66	-0.0011	-58.965	4.062

FMIN,FMAX
?0,0

Figure 33 (Continued)

```
NØ
?9
 512 POINT TRANSFORM OF  59 STAGE  1Ø BIT FILTER

FMIN,FMAX
?1.32,1.37
 85        Ø.9419        -Ø.52Ø       1.312
 86        Ø.923Ø        -Ø.696       1.328
 87        Ø.9ØØ4        -Ø.911       1.344
 88        Ø.8742        -1.168       1.359
 89        Ø.8443        -1.47Ø       1.375
FMIN,FMAX
?1.76,1.8
113        Ø.Ø119       -38.472       1.75Ø
114        Ø.ØØ65       -43.8Ø7       1.766
115        Ø.ØØ27       -51.46Ø       1.781
116        Ø.ØØØ3       -72.ØØ9       1.797
117       -Ø.ØØ11       -58.951       1.812
FMIN,FMAX
?Ø,Ø
NØ
?Ø
```

Figure 33 (Concluded)

```
C,L,A
?.5,10,.45
     MAXIMUM TAP =     511
FS,FC,M
?16,4,10
    303       0     -55       0       7       0       0       0       0       0
SUM=   1021  SSQ=    1.727  SRS=    1.314
NØ
?7
 128 POINT TRANSFORM OF  21 STAGE  10 BIT FILTER

FMIN,FMAX
?1,7
  9        1.0014          0.012      1.000
 10        1.0017          0.015      1.125
 11        1.0018          0.016      1.250
 12        1.0015          0.013      1.375
 13        1.0006          0.005      1.500
 14        0.9988         -0.010      1.625
 15        0.9960         -0.035      1.750
 16        0.9917         -0.072      1.875
 17        0.9857         -0.125      2.000
 18        0.9776         -0.197      2.125
 19        0.9671         -0.290      2.250
 20        0.9539         -0.410      2.375
 21        0.9377         -0.559      2.500
 22        0.9182         -0.741      2.625
 23        0.8953         -0.960      2.750
 24        0.8689         -1.220      2.875
 25        0.8390         -1.525      3.000
 26        0.8056         -1.877      3.125
 27        0.7690         -2.282      3.250
 28        0.7293         -2.742      3.375
 89        0.6869         -3.263      3.500
 30        0.6422         -3.846      3.625
```

Figure 34: 2 to 1 Resampling Filter, r = 16KHz

31	0.5958	-4.498	3.750
32	0.5482	-5.221	3.875
33	0.5000	-6.021	4.000
34	0.4518	-6.901	4.125
35	0.4042	-7.869	4.250
36	0.3578	-8.928	4.375
37	0.3131	-10.085	4.500
38	0.2707	-11.349	4.625
39	0.2310	-12.726	4.750
40	0.1944	-14.227	4.875
41	0.1610	-15.864	5.000
42	0.1311	-17.650	5.125
43	0.1047	-19.603	5.250
44	0.0818	-21.745	5.375
45	0.0623	-24.105	5.500
46	0.0461	-26.724	5.625
47	0.0329	-29.660	5.750
48	0.0224	-33.000	5.875
49	0.0143	-36.895	6.000
50	0.0083	-41.637	6.125
51	0.0040	-47.935	6.250
52	0.0012	-58.740	6.375
53	-0.0006	-64.633	6.500
54	-0.0015	-56.523	6.625
55	-0.0018	-54.891	6.750
56	-0.0017	-55.329	6.875
57	-0.0014	-57.154	7.000
58	-0.0009	-60.452	7.125

Figure 34 (Continued)

```
2750    Q = Q*cos(3.3*PII/FS)
7550    Y = 2.*Y.
```

The corresponding printout (at a 16KHz sampling rate) is shown in figure 35.

3. DIGITAL DIFFERENTIATION

In Reference 1, the digital Hilbert transform was derived from the sampling theorem,

$$x(t) = \sum_{k=-\infty}^{\infty} x_k \frac{\sin(\pi rt - k\pi)}{(\pi rt - k\pi)} \qquad (1)$$

$$\sum_{k=-\infty}^{\infty} (-1)^k x_k \frac{\sin \pi rt}{\pi(rt-k)} ,$$

by taking the Hilbert transform of (1) and sampling the result at the rate r; $\hat{x}_n = \hat{x}(n/r)$. The result was

$$\hat{x}_n = \frac{2}{\pi} \sum_{k=-\infty}^{\infty} \frac{x_{n-k}}{k} , \text{ k odd.} \qquad (2)$$

Similarly, we can differentiate (1),

$$\dot{x}(t) = \sum_{k=-\infty}^{\infty} (-1)^k x_k r \left[\frac{\cos \pi rt}{rt-k} - \frac{\sin \pi rt}{\pi(rt-k)^2} \right] , \qquad (3)$$

and, by sampling this result at the rate r, we obtain the digital derivative

$$\dot{x}_n = \dot{x}(n/r) = r \sum_{\substack{k=-\infty \\ k \neq n}}^{\infty} (-1)^{n-k} \frac{x_k}{n-k} . \qquad (4)$$

The preceeding results could also have been obtained by deriving the appropriate filter frequency response, H(f), and finding the corresponding unit response

C,L,A
?.5,10,0
MAXIMUM TAP = 511
FS,FC,M
?16,1.45,60

386	110	-107	-144	-51	29	16	-48	[illegible]	-46
-1	3	-30	-52	-38	-9	0	-18	[illegible]5	-29
-9	0	-9	-22	-20	-7	1	-3	12	-12
-4	2	1	-5	-6	-2	2	2	-1	-3
-1	2	2	0	-1	0	1	1	1	0
0	0	1	0	0	0	0	0	0	0

SUM= 25 SSQ= 2.658 SRS= 1.630
NO
?7
128 POINT TRANSFORM OF 121 STAGE 10 BIT FILTER

FMIN,FMAX
?0,4

1	0.0089	-35.023	0.000
2	0.1169	-12.619	0.125
3	0.3417	-3.306	0.250
4	0.4844	-0.276	0.375
5	0.5040	0.069	0.500
6	0.4996	-0.003	0.[illegible]25
7	0.4997	-0.005	0.750
8	0.5009	0.015	0.875
9	0.5018	0.031	1.000
10	0.4994	-0.010	1.125
11	0.5001	0.002	1.250
12	0.5007	0.013	1.375
13	0.5008	0.015	1.500
14	0.5002	0.004	1.625
15	0.4993	-0.012	1.750
16	0.4999	-0.001	1.875
17	0.5002	0.003	2.000
18	0.4995	-0.009	2.125
19	0.5003	0.006	2.250
20	0.5003	0.006	2.375
21	0.4998	-0.003	2.500

Figure 35: Double Sideband Filter, r = 16KHz

22	0.5002	0.004	2.625
23	0.5008	0.014	2.750
24	0.5017	0.030	2.875
25	0.4158	-1.602	3.000
26	0.2020	-7.873	3.125
27	0.0326	-23.728	3.250
28	-0.0042	-41.566	3.375
29	0.0007	-57.526	3.500
30	0.0012	-52.740	3.625
31	0.0010	-54.099	3.750
32	0.0002	-66.546	3.875
33	-0.0011	-53.440	4.000
34	0.0011	-52.831	4.125

FMIN,FMAX
?4,8

33	-0.0011	-53.440	4.000
34	0.0011	-52.831	4.125
35	0.0012	-52.469	4.250
36	-0.0013	-51.912	4.375
37	0.0020	-47.759	4.500
38	0.0000	-83.942	4.625
39	-0.0012	-52.186	4.750
40	0.0009	-55.039	4.875
41	-0.0014	-51.233	5.000
42	-0.0007	-56.666	5.125
43	-0.0000	-8[illegible].310	5.250
44	0.0008	-55.991	5.375
45	0.0009	-54.975	5.500
46	0.0007	-56.778	5.625
47	0.0021	-47.717	5.750
48	-0.0001	-73.318	5.875
49	-0.0015	-50.712	6.000
50	-0.0009	-54.846	6.125

Figure 35 (Continued)

```
 51        0.0011        -52.871       6.250
 52        0.0006        -58.322       6.375
 53       -0.0014        -51.080       6.500
 54       -0.0006        -58.319       6.625
 55       -0.0006        -58.677       6.750
 56        0.0011        -53.219       6.875
 57       -0.0011        -53.119       7.000
 58       -0.0024        -46.345       7.125
 59       -0.0004        -62.737       7.250
 60       -0.0018        -48.808       7.375
 61       -0.0001        -72.573       7.500
 62       -0.0006        -57.964       7.625
 63       -0.0004        -62.600       7.750
 64        0.0002        -68.440       7.875
 65       -0.0025        -46.080       8.000
 66        0.0002        -68.440       8.125
FMIN,FMAX
?0,0
NO
?9
 512 POINT TRANSFORM OF 121 STAGE  10 BIT FILTER

FMIN,FMAX
?.25,.37
  9        0.3417         -3.306       0.250
 10        0.3911         -2.134       0.281
 11        0.4319         -1.272       0.313
 12        0.4630         -0.668       0.344
 13        0.4844         -0.276       0.375
FMIN,FMAX
?0,0
NO
?0
```

Figure 35 (Concluded)

$$h_n = \int_{r/2}^{-r/2} H(f) \epsilon^{j2\pi nf/r} df. \tag{5}$$

If

$$H(f) = j \operatorname{sgn} f \tag{6}$$

we obtain

$$h_n = \begin{cases} \frac{2}{\pi} \frac{1}{n}, & n \text{ odd} \\ 0, & n \text{ even} \end{cases} \tag{7}$$

for the Hilbert transforming filter. If

$$H(f) = j2\pi f \tag{8}$$

we obtain

$$h_n = \begin{cases} r \frac{(-1)^n}{n}, & n \neq 0 \\ 0, & n = 0 \end{cases} \tag{9}$$

for the differentiating filter. Equation (9) was used with a Hanning window for differentiation in the CONFIDE program. Note that equation (4) can also be written

$$\dot{x}_n = r \sum_{\substack{k = -\infty \\ k \neq 0}}^{\infty} (-1)^k \frac{x_{n-k}}{k} \tag{10}$$

Rabiner and Steiglitz (25) have pointed out that the above is not the thing to do. They note that the magnitude and phase response corresponding to equation (8) is hard to realize digitally because of the phase discontinuity shown in figure 36c. By introducing a half-sample delay,

$$H_1(f) = j2\pi f \; e^{-j\pi f/r}, \quad |f| < \frac{r}{2} \tag{11}$$

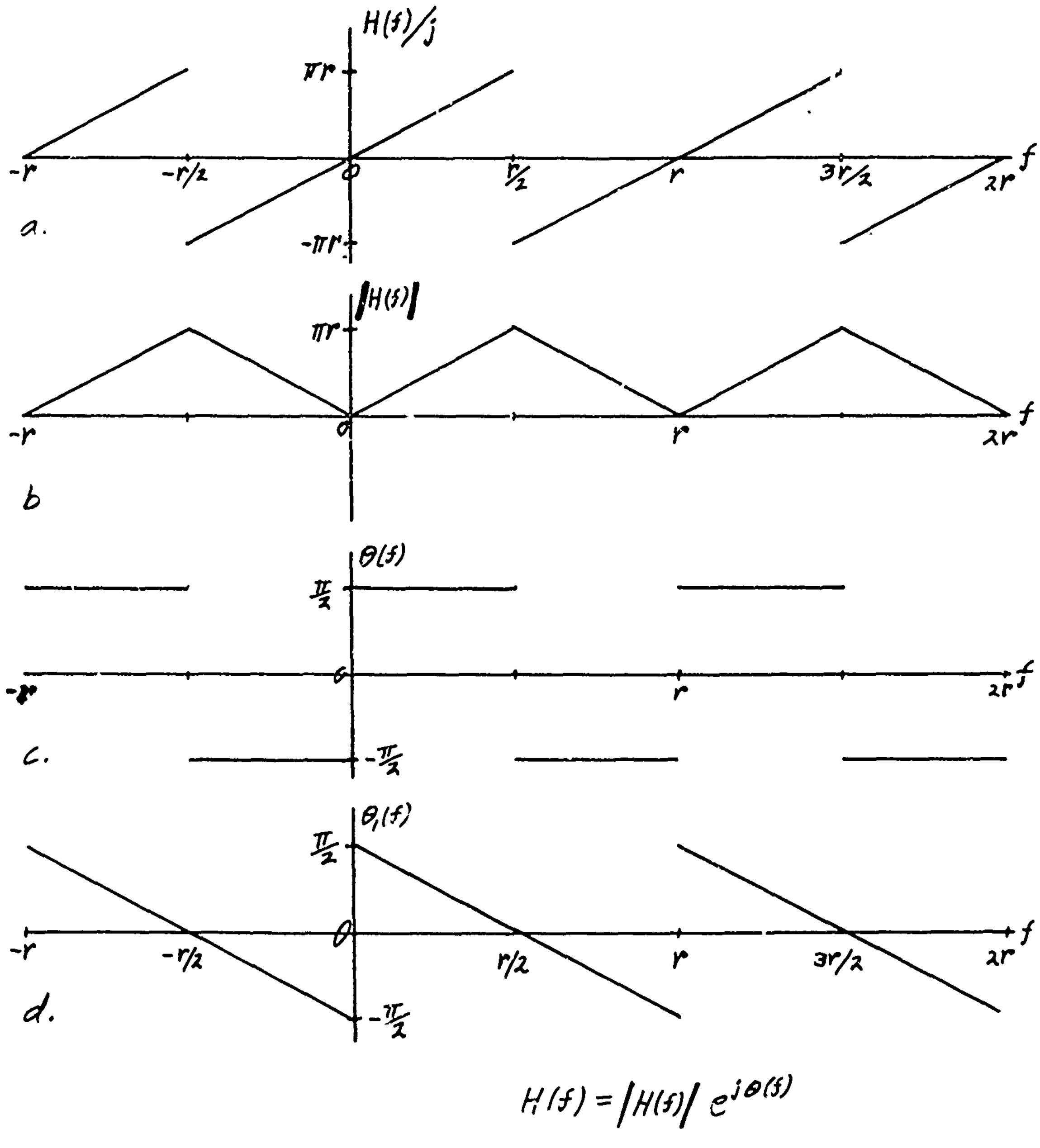


Figure 36: Frequency Response of an Ideal Wideband Differentiator

the phase response of figure 36d is obtained, whereby the phase discontinuities occur only at frequencies at which $|H(f)| = 0$.

From equation (3) using $t = (n-1/2)/r$ instead of $t = n/r$, we obtain

$$\dot{x}_n = \dot{x}\left(\frac{n-1/2}{r}\right) = \frac{r}{\pi} \sum_{k=-\infty}^{\infty} (-1)^{n-k} \frac{x_k}{(n-k-\frac{1}{2})^2} \tag{12}$$

or

$$\dot{x}_n = \frac{r}{\pi} \sum_{k=-\infty}^{\infty} (-1)^k \frac{x_{n-k}}{(k-\frac{1}{2})^2} = \frac{4r}{\pi} \sum_{k=-\infty}^{\infty} (-1)^k \frac{x_{n-k}}{(2k-1)^2} . \tag{13}$$

Alternatively we could obtain

$$h_n = \frac{4r}{\pi} \frac{(-1)^n}{(2n-1)^2} \tag{14}$$

from equations (5) and (11). Equations (9) and (14) are compared in figure 37, for $r = 1$. Equation (14) is seen to decay quadradically and to more closely approximate a simple $(x_i - x_{i-1}) \cdot r$ "differentiator". The unit response has to be truncated to finite duration. Rabiner and Steiglitz[25] present excellent results for the frequency sample specification method. However, any of the methods of Section III may be used.

Another way of viewing these results is as follows. A filter with impulse response

$$\frac{\sin \pi r t}{\pi r t} \tag{15}$$

would pass a signal bandlimited to $-r/2<f<r/2$, undistorted. The impulse response of the corresponding differentiator is given by the derivative of (15):

$$h(t) = \frac{1}{t}\left[\cos \pi r t - \frac{\sin \pi r t}{\pi r t}\right] . \tag{16}$$

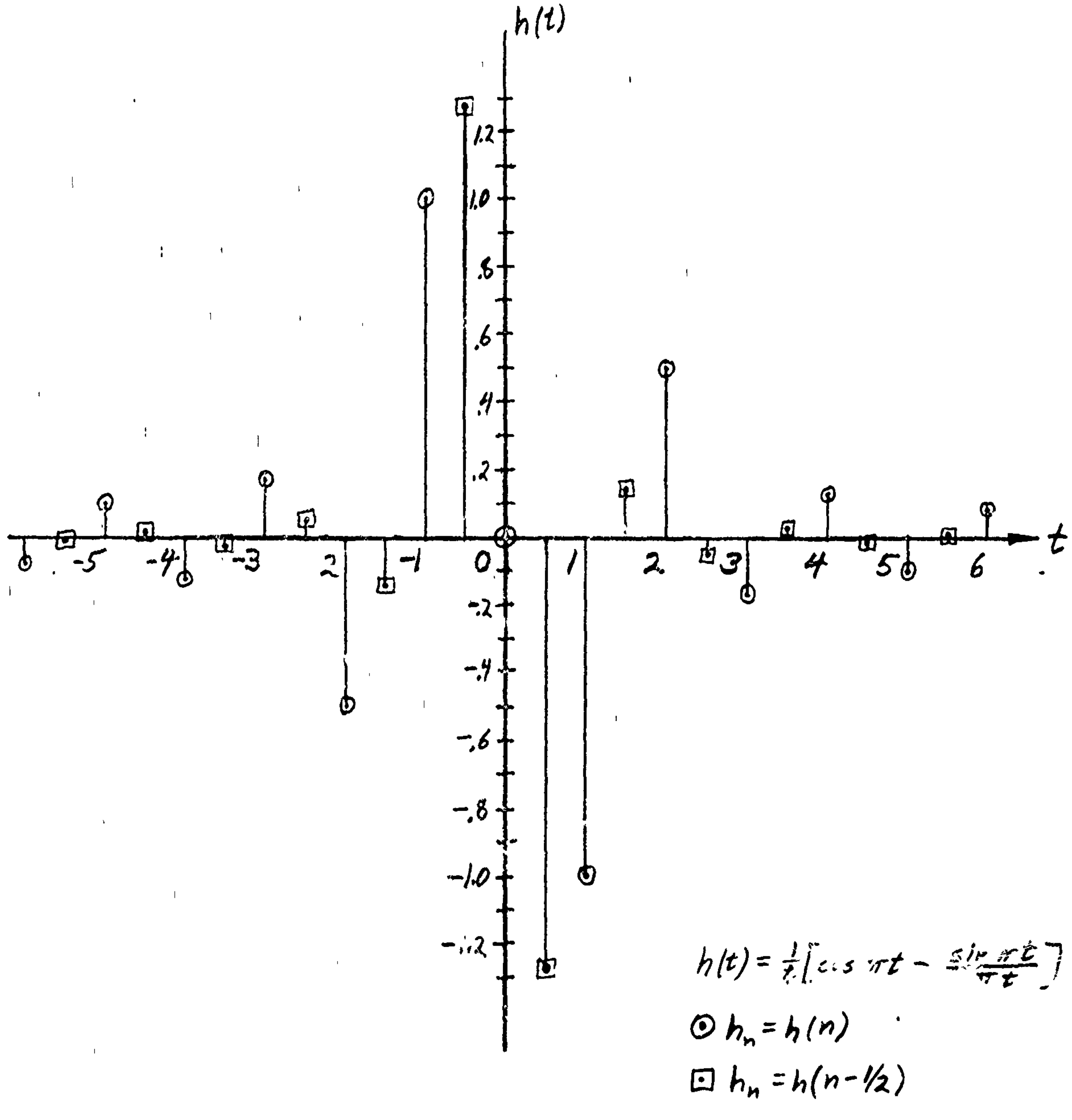


Figure 37: Unit Response of Two Wideband Differentiators

Equation (15) can be thought of as a bandlimited inpulse $\delta(t)$, and (16) as a bandlimited doublet, $\dot{\delta}(t)$. The unit response of a digital differentiator is obtained by sampling (16) at the rate r. The usual wideband differentiator [5,6,11,26,27], such as the one used in our simulations, samples (16) at the zeros of the sine term. Obviously, a shorter response is obtained by sampling at the zeros of the cosine. In fact, the latter approach indicates that the simple-minded two-term differentiator, such as the one used in our breadboard, is a reasonable first approximation.

SECTION VII

COMPUTER SIMULATION

Simulations were made for three modulation schemes. Single sideband (SSB), double sideband (DSB), and frequency modulation (FM). The simulations were initially programed on the Burroughs 5500 time share system. The working programs were then transfered to the Philco 212 computer system located at Willow Grove. Signal to noise runs were made on the Philco 212.

The calculations of the tap weights for the convolutional filters were performed on the Burroughs 5500 time share system. The program CONFIDE calculated the tap weights to the desired bit accuracy and then performed a Fourier Transform on the weights. The results of the Fourier Transform were printed out in order that the frequency selective properties of the filter could be checked.

The various modulation system simulations were carried out on a low pass equivalent basis for the DSB, SSB, and FM systems. All filtering was done with convolutional digital filters with complex signals where appropriate. The basic sampling rate for input and output of all simulations was 15KHz. This sampling rate was thought to be a compromise between the sampling rate needed for DSB and SSB and the rate needed for FM. Based on the results of these initial simulations an effective sampling rate for input and output of 8KHz was chosen for the breadboard.

The SSB and DSB systems were tested by passing a pulse derived from a 4KHz 50 o/o raised cosine spectrum through the system. The spectrum of the test pulse is shown in figure 16b of Section III. 6 (B = 4KHz). This pulse had eleven (11) non-zero terms in it after rounding to 7 bit precision. The reason for using the above mentioned pulse was to simulate the inherent filtering a real input to the systems would experience. Throughout the simulations the real channel data stream is designated as XR_ and the imaginary channel data stream as XI_. Whenever possible the output of both real and imaginary channels were saved as an aid to checking the output of the filters as the signal progressed through the system. The simulations can best be understood by relating the programs for the various simulations to their block diagrams (figures 38, 39 and 40). All of the programs are documented with comment cards and should be reasonably easy to understand by anyone with a good working knowledge of Fortran and some background in digital communication system design.

The noise added to the input to the receiver was derived from a

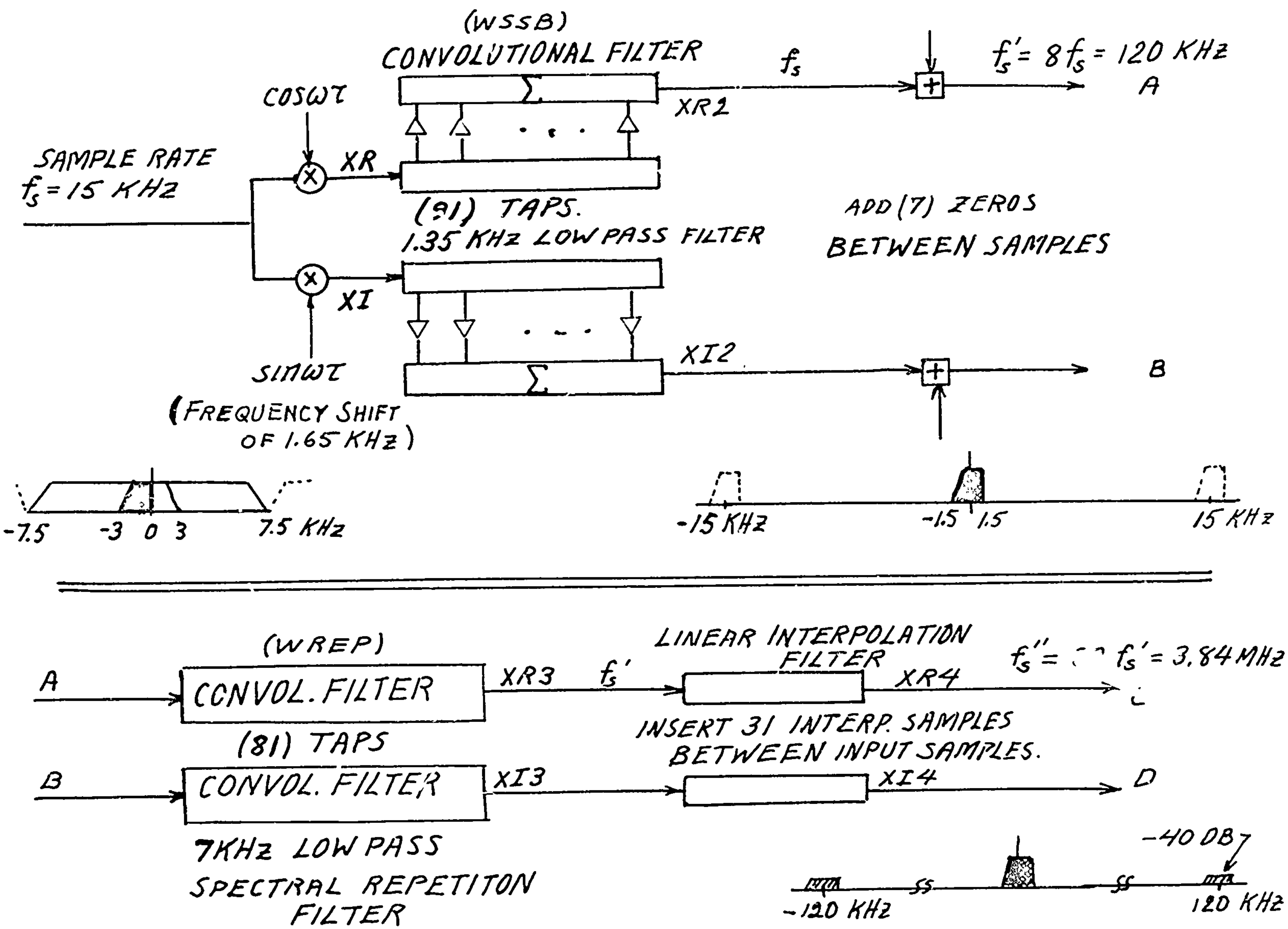


Figure 38: S.S.B. System Simulation

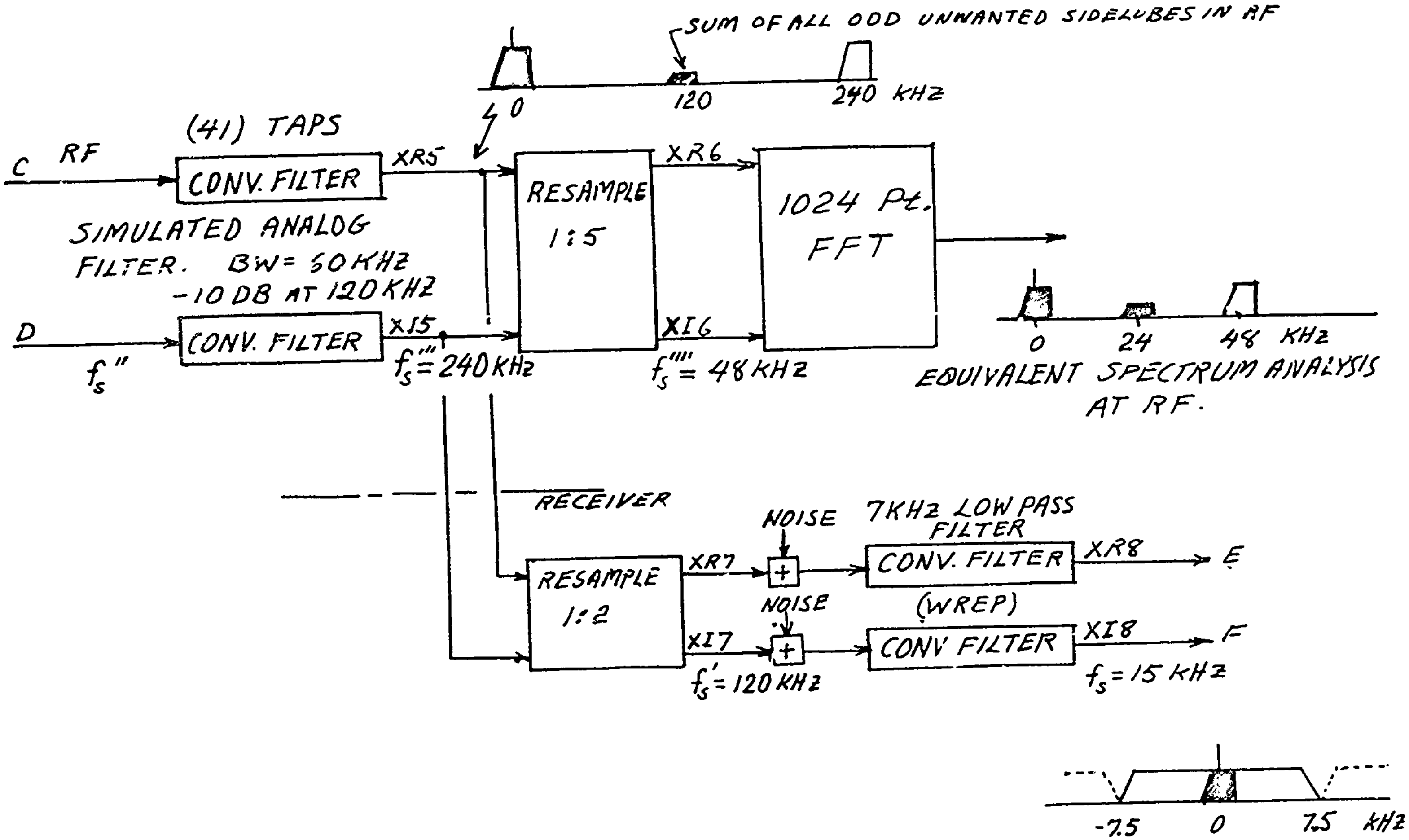


Figure 38 (Continued)

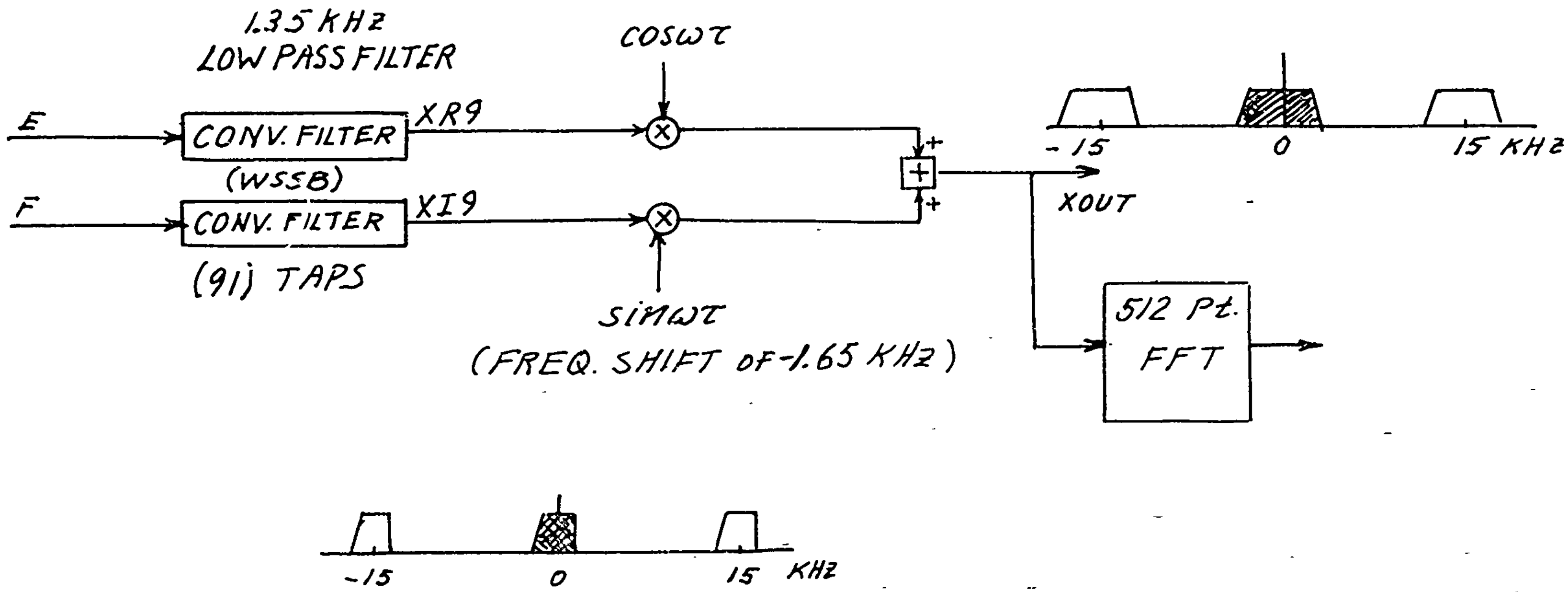


Figure 38 (Concluded)

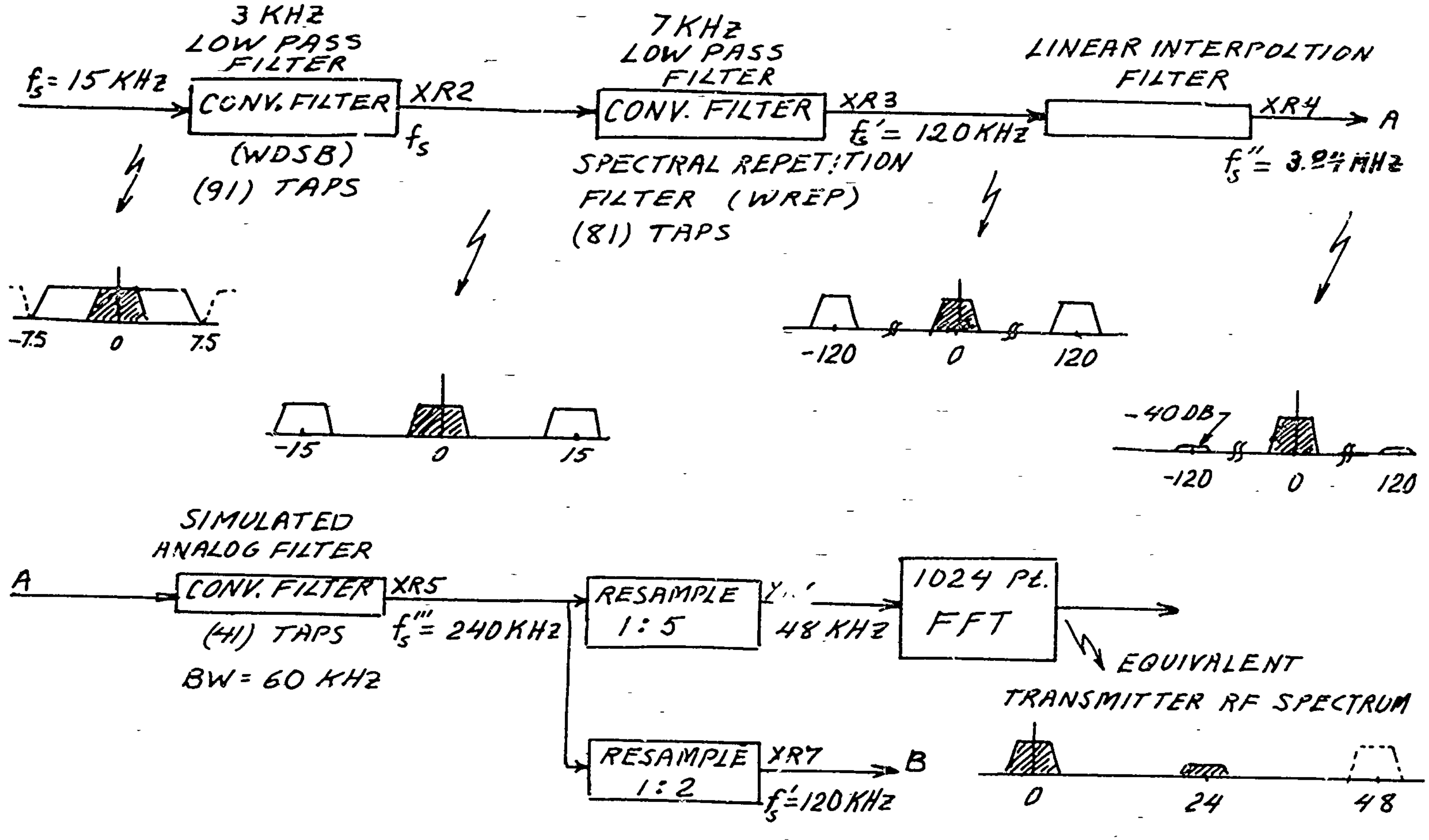


Figure 39: DSB System Simulation

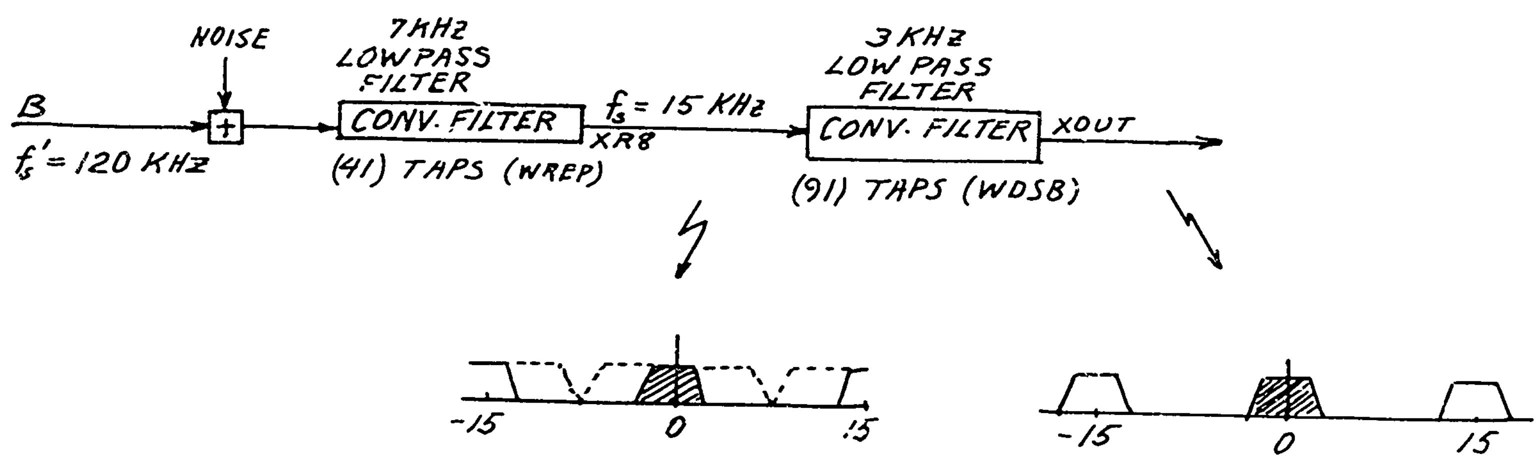


Figure 39 (Continued)

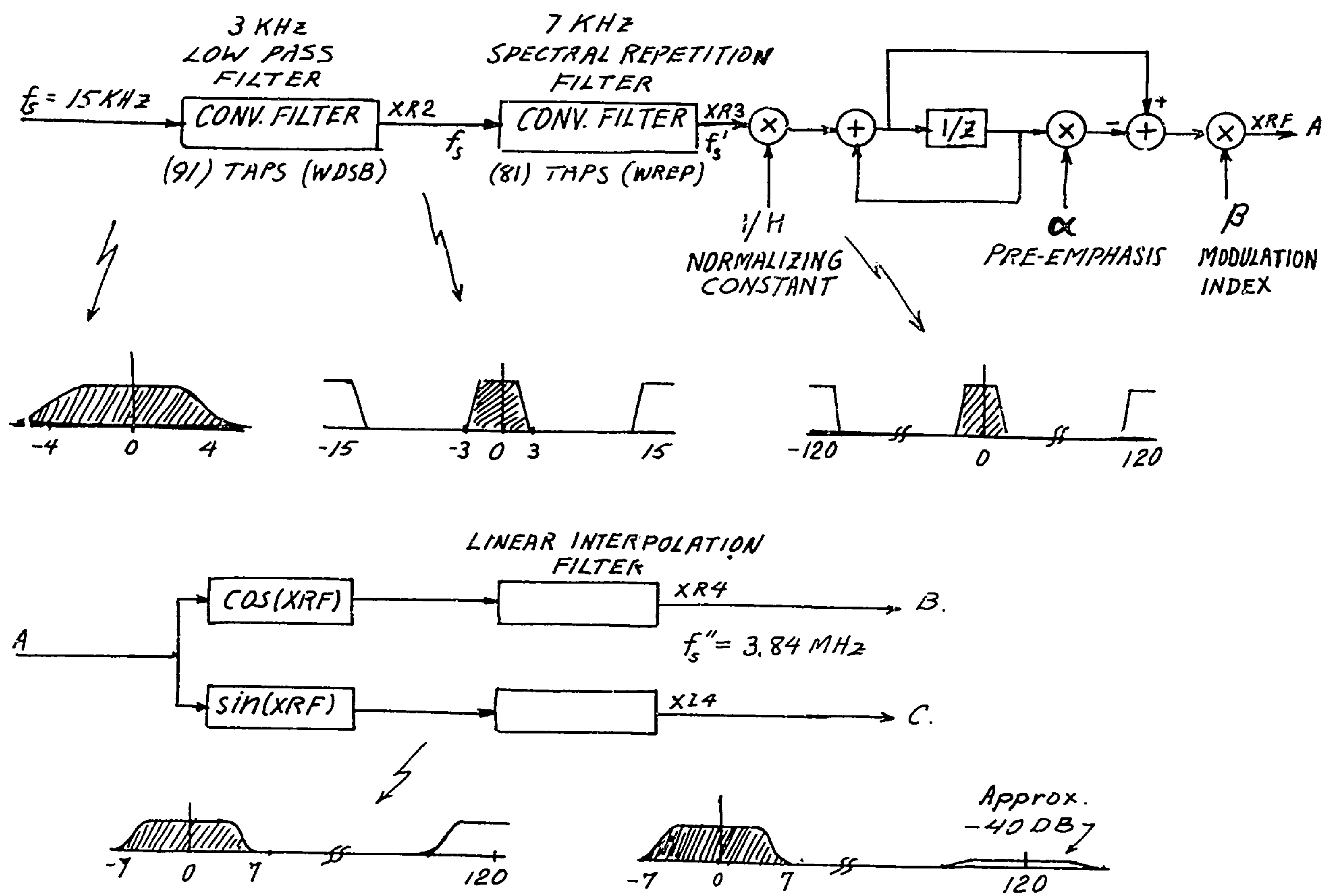


Figure 40: FM and PM System Simulation

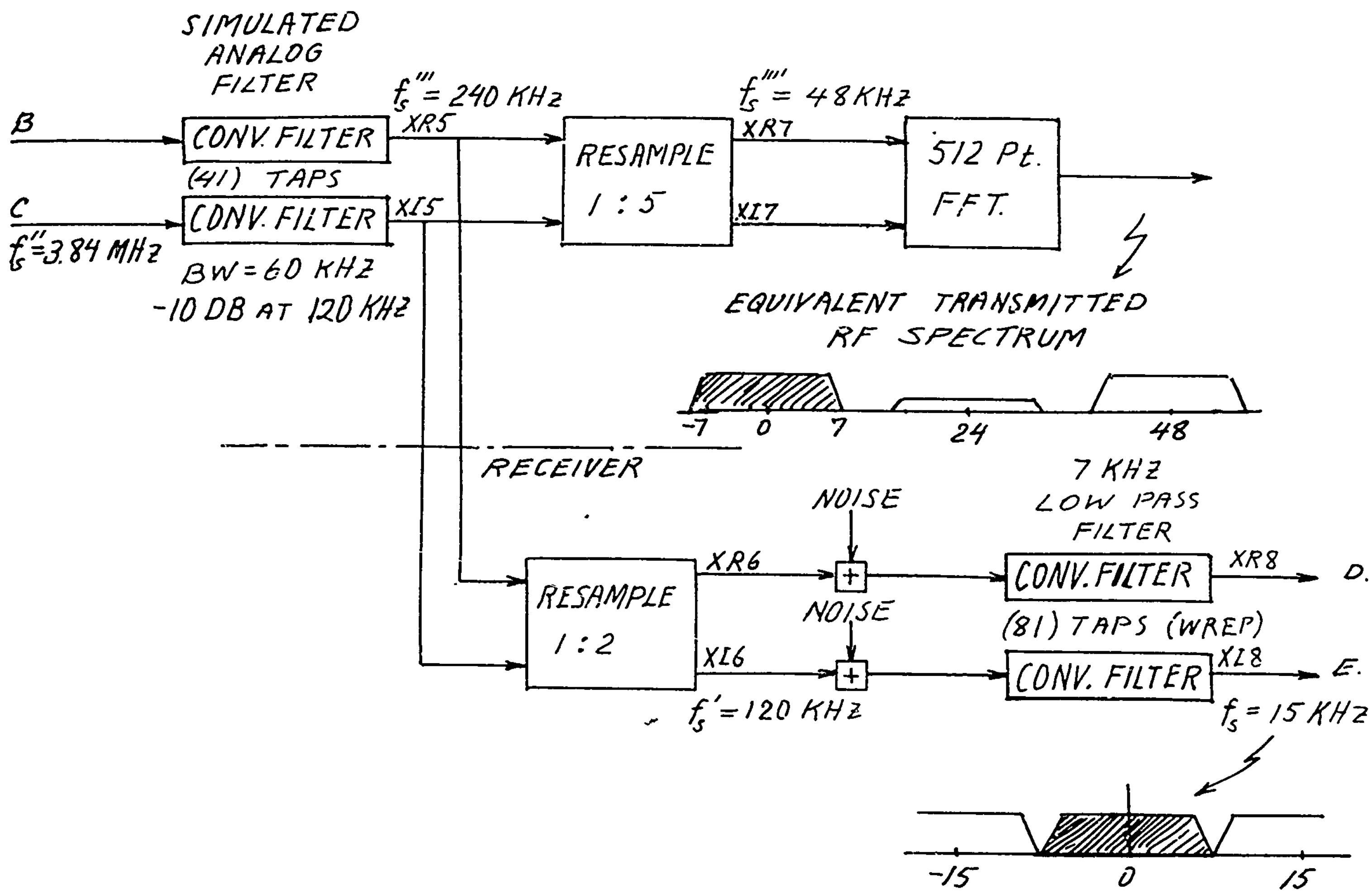


Figure 40 (Continued)

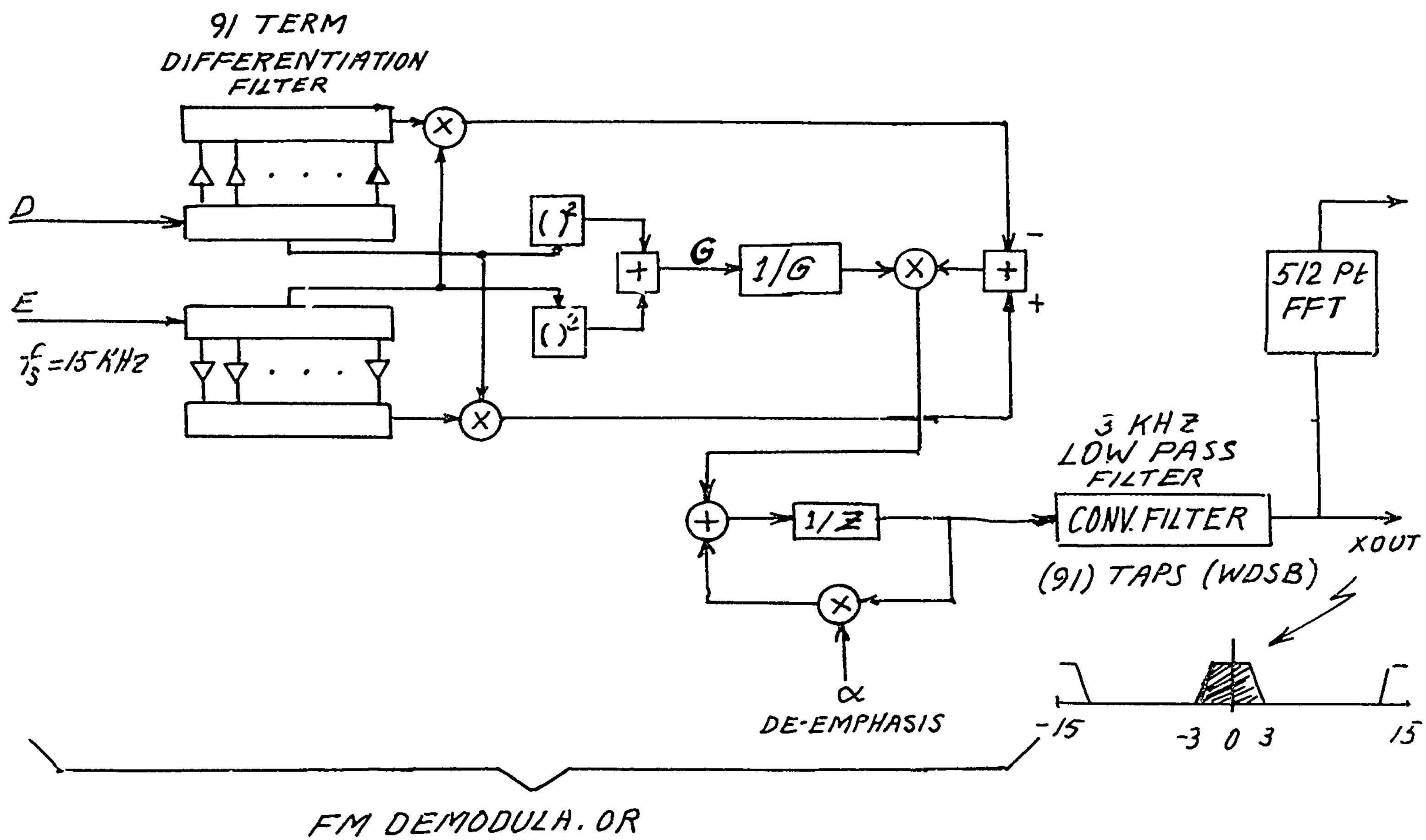


Figure 40 (Concluded)

gaussian random number generator and the specific noise power at that point was calculated. The noise was added at a level to compensate for the bandwidth of the receiver. Signal-to-noise curves were run for the DSB and SSB cases and plotted in figures 41 and 42. The method used to calculate the signal-to-noise ratio out of the receiver was that the spectrum of the output wave form with no noise (-100dB) added to the input to the receiver was calculated. The difference between this spectrum and the output spectrum with noise added to the receiver was calculated using the no noise spectrum as a reference. Also the distortion of the spectrum of the signal from the input of the systems to the output were calculated to obtain a feel for the amount of distortion the systems are adding due to the number of bits of precision that were carried. These figures are:

DSB	23.8 DB
SSB	21.2 DB

These signal to distortion ratios are not particularly good indicates that the signals should be carried through the system at greater than 8 bit precision. And the filter should be implemented with more than 10 bit precision. Good precision for the signal would be 10-12 bits and 12-14 bits for the filters. Actually, 10-12 bits could be used for the filters provided the number of taps is kept low by lowering the sampling rate.

The FM system was checked using a longer pulse train, about 20 pulses. This pulse train was passed through the system. The pulse train was FM modulated filtered and translated to 3.34MHz resampled and demodulated and printed out. As can be seen in figure 43 the pulse was faithfully reproduced at both a height of 1.0 and 0.5. These two illustrations show that the digital FM system simulated is feasible. As before the signal was carried to 8 bit precision and the filtering was implemented to 10 bit precision. Provision was made in the simulation for preemphasis. The parameter α in figure 40 may vary from 0 (no preemphasis) to 1 (phase modulation). This parameter was not investigated.

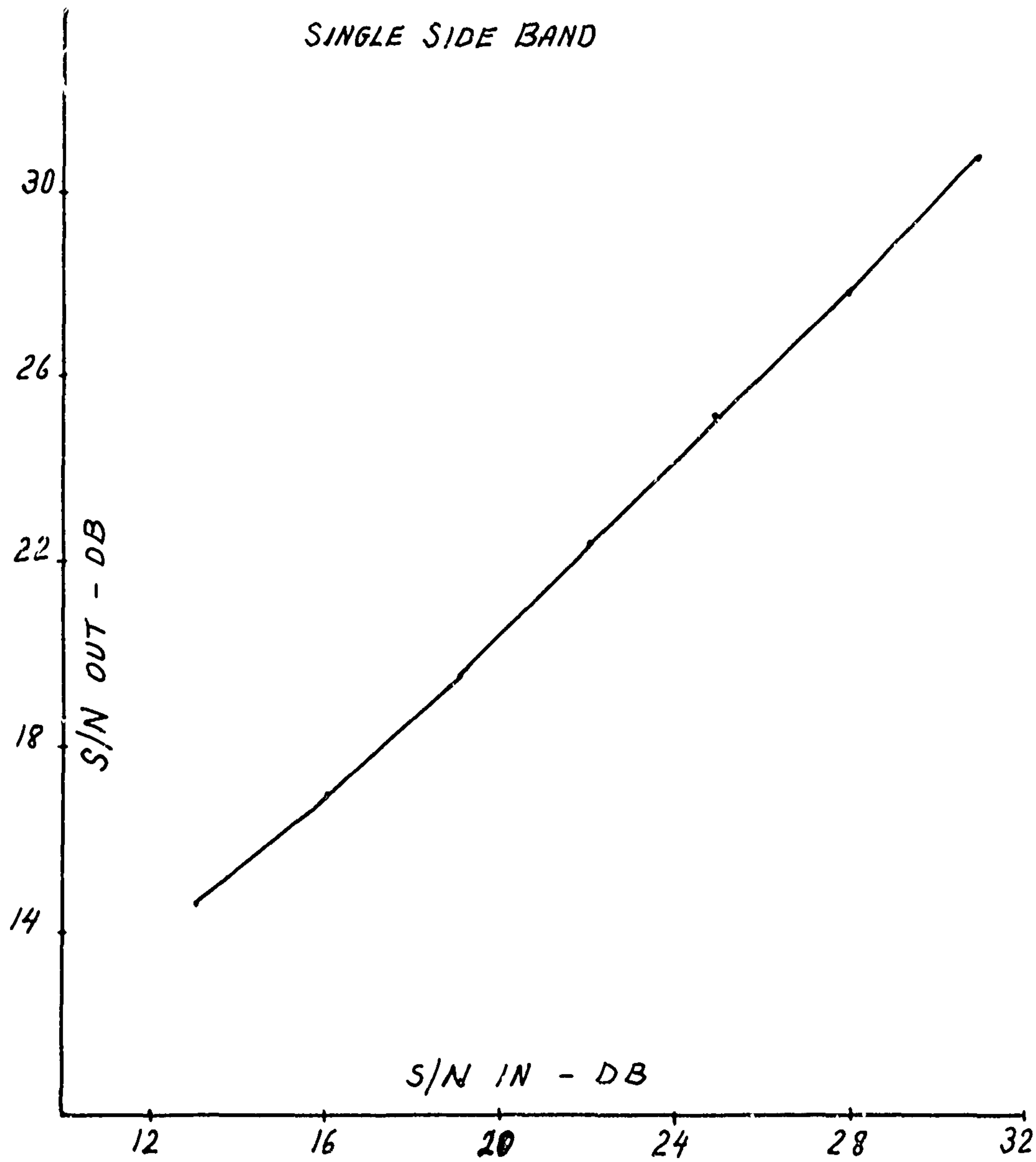
1. SUBROUTINE DESCRIPTION

GAUSS (S, SQH, SGH)

This subroutine returns a gaussian random variable X (in volts) with zero mean and standard deviation $SQH(10)^{-SGN/20}$. SQH is the RMS value of the signal where noise is added and SGN is the signal to noise ratio.

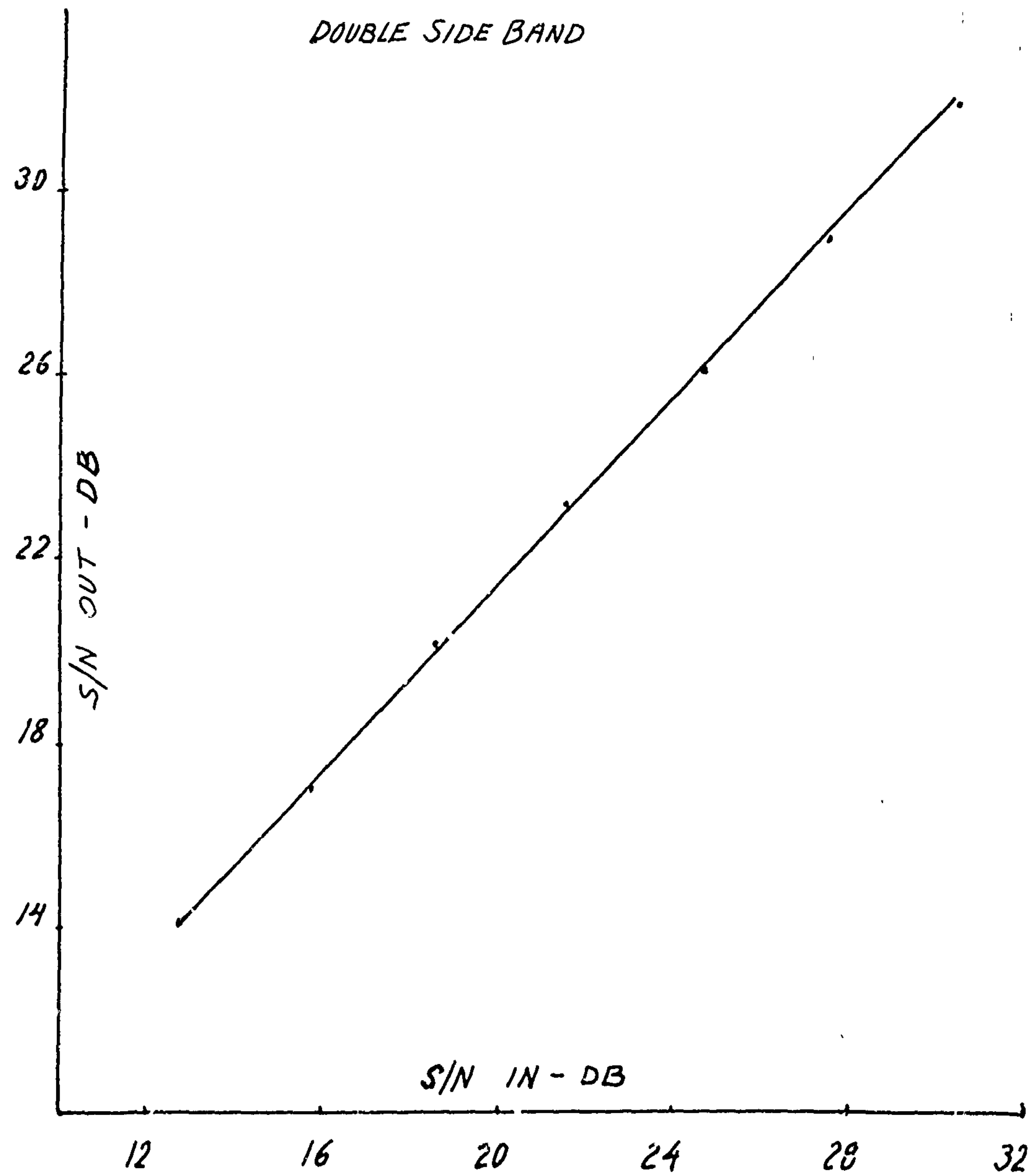
RND (X, XMAX, L)

This subroutine rounds X to L bit precision. XMAX is the maximum magnitude of the variable X.



SIGNAL-TO-DISTORTION RATIO
THROUGH SYSTEM = 21.2 DB.

Figure 41: SSB Simulation Results



SIGNAL-TO DISTORTION
RATIO THROUGH SYSTEM = 23.85 db.

Figure 42. DSB Simulation Results

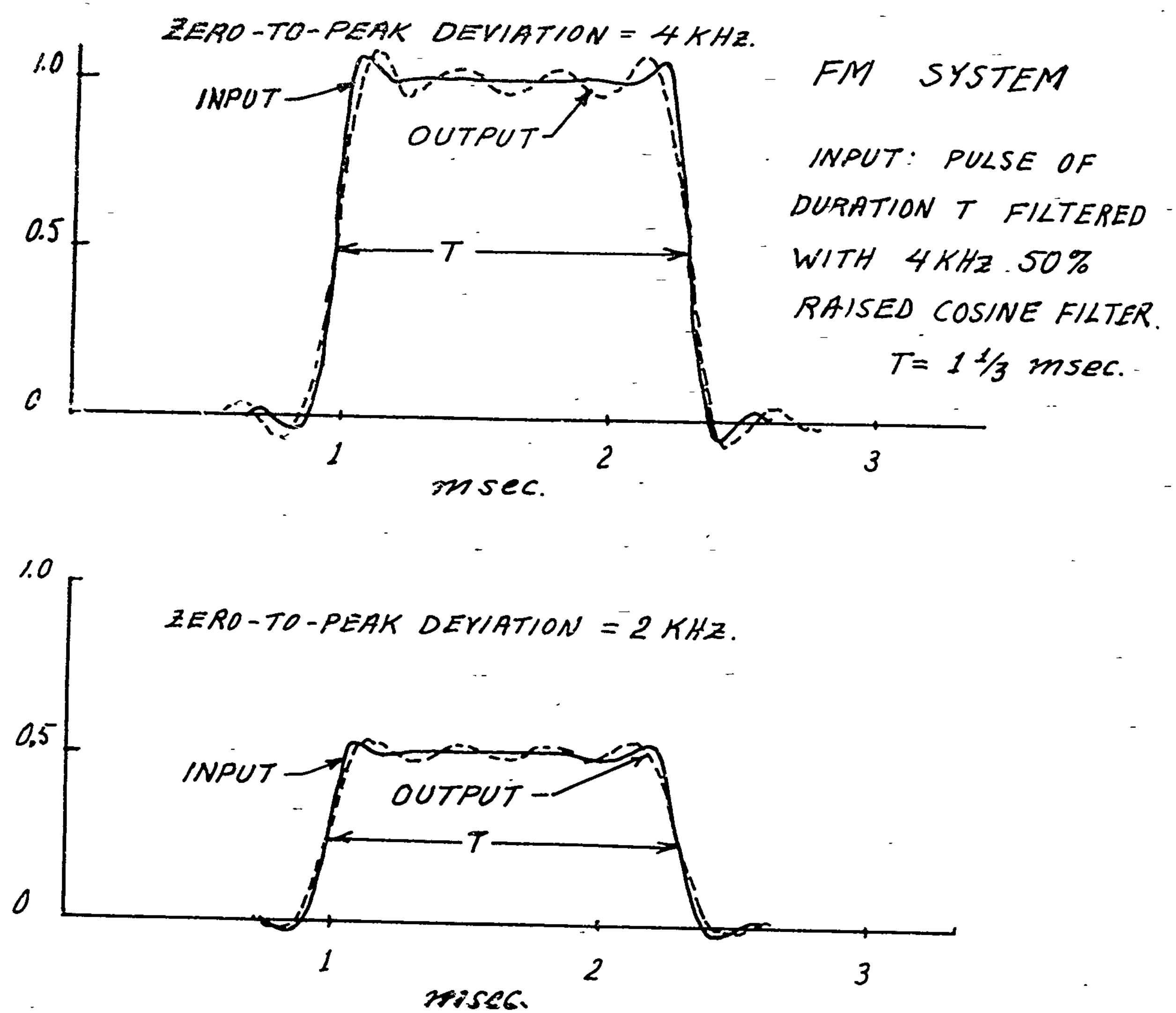


Figure 43: FM Simulation Results

CS (X, L)

Computes cosine X to L bit precision.

SN (X, L)

Computes sine X to L bit precesion.

FFT (X, NSTAGE, SIGN)

Subroutine to take Fourier Transform. For details see comments in listing of the subroutine.

ERR(XR, XI, IL, M, FM, FT)

This subroutine computes the normalized magnitude of the spectrum of a complex signal. XR is the real part of the signal, XI is the imaginary part of the signal, 2^{IL} is the size of the transform used, M is the number of complex samples of input data, FM is the output magnitude of the spectrum and FT is a complex working matrix.

FIND (XR, XI, IL, M, REP, FT)

This subroutine calculates and prints out the normalized spectrum of a complex wave form with XR the real part and XI the imaginary part. All parameters are the same as in ERR except that REP is the rate at which the time wave form is sampled. This subroutine calls FFT.

2 SSB SIMULATION

```
C SINGLE SIDEBAND SIMULATION  D.L.FLETCHER
      DIMENSION WSSE(100),WREP(100),ANLCG(100),X(100),XR(300),
     *XI(300),XR2(400),XI2(400),XR3(850),XI3(850),XR4(150),XI4(150),
     *XR5(1650),XI5(1650),XR6(350),XI6(350),XR7(900),XI7(900),
     *XR8(300),XI8(300),XR9(300),XI9(300),XCUT(300),FTM(512)
      DIMENSION F1(512),FM(1024),F2(512)
      COMPLEX FT(2,1024),C
      DATA A1,A3,A4,A8,A9,A0/1.74,3.56,3.56,20.03,34.86,34.86/
      DATA LF,L/10,8/
      DATA MSIG,MS,MR,MA/5,45,40,20/
      DATA NS,NSB,NRF,NA/11,91,81,41/
C NS=NO. OF TERMS IN INPUT SIGNAL = 2*MSIG+1
C NSB=NO. OF TERMS IN SSB FILTER = 2*MS+1
C NRF=NO. OF TERMS IN REP FILTER = 2*MR+1
C NA=NO. OF TERMS IN ANALOG FILTER = 2*MA+1
      DATA FZS,FZR,FZA/.1893,.1258,.0312/
      DATA SGN /15./
      PI=3.14159265
CCALCULATE TAP WEIGHTS
      DO 50 I=1,MS
      FI=I
      WIN=.54+.46*COS(PI*FI/50.)
      IMS=MS+1+I
      WSSE(IMS)=WIN*SIN(FZS*PI*FI)/(PI*FI*FZS)
      CALL RND(WSSE(IMS),1.0,LF)
      JMS=MS+1-I
   50 WSSE(JMS)=WSSE(IMS)
      WSSE(MS+1)=1.0
      CALL RND(WSSE(MS+1),1.0,LF)
      FMR=MR
      DO 60 I=1,MR
      FI=I
      AT=FI*FI/FMR
      WIN=.54+.46*COS(AT)
      IMR=MR+I+1
      WREF(IMR)=WIN*SIN(FZR*PI*FI)/(PI*FI*FZR)
      CALL RND(WREF(IMR),1.0,LF)
      JMR=MR+1-I
   60 WREF(JMR)=WREF(IMR)
      WREF(MR+1)=1.0
      CALL RND(WREF(MR+1),1.0,LF)
      FMA=MA
      DO 70 I=1,MA
      FI=I
      AT=FI*FI/FMA
      WIN=.54+.46*COS(AT)
      IMA=MA+1+I
      ANLCG(IMA)=WIN*SIN(FZA*PI*FI)/(PI*FI*FZA)
      JMA=MA+1-I
      ANLCG(JMA)=ANLCG(IMA)
70    CONTINUE
      ANLCG(MA+1)=1.0
C LOAD 50 PERCENT RC PULSE
      A=8.0*PI/15.0
      DO 80 I=1,MSIG
      FI=I
```

```
      IMSIG=MSIG+1+I
      X(IMSIG)=SIN(A*FI)*COS(.5*A*FI)/(A*FI*(1.-(8.*FI/15.)**2))
      CALL RND(X(IMSIG),1.0,7)
      JMSIG=MSIG+1-I
   80 X(JMSIG)=X(IMSIG)
      X(MSIG+1)=1.0
      CALL RND(X(MSIG+1),1.0,7)
      PRINT 81,(X(I),I=1,NS)
81    FORMAT(1H1,17HINPUT WAVE FORM   /(10F10.5))
CSHIFT SPECTRUM BY 1.65 KC
      W=1.65/15.0
      DO 110 I=1,NS
      FI=I-MSIG-1
      XW=2.0*PI*W*FI
      XR(I)=X(I)*CS(XW,LF)
      CALL RND(XR(I),1.0,L)
      XI(I)=X(I)*SN(XW,LF)
110   CALL RND(XI(I),1.0,L)
C CONVOLVE WITH FIRST FILTER
      DO 112 I=1,NS
C SHIFT ALL TERMS BY NSB
      IN=NSB+NS-I
      IS=NS+1-I
      XR(IN)=XR(IS)
 112  XI(IN)=XI(IS)
      NS1=NSB-1
      DO 113 I=1,NS1
      XR(I)=0.0
113   XI(I)=0.0
C NF=NO. OF TERMS OUT OF FIRST FILTER
      NF=NSB+NS-1
      DO 115 I=1,NF
      DO 114 J=1,NSB
      IJ=I+J-1
      XR2(I)=XR2(I)+WSSB(J)*XR(IJ)
 114  XI2(I)=XI2(I)+WSSB(J)*XI(IJ)
      CALL RND(XR2(I),A1,L)
115   CALL RND(XI2(I),A1,L)
      CALL ERR(XR2,XI2,9,NF,FM,FT)
      PRINT 10
10    FORMAT(1H0,3HWSB )
      CALL FTAD(XR2,XI2,9,NF,15000.,FT)
CEFFECTIVELY FILL WITH ZEROS AND CONVOLVE WITH 7 KC LOW-PASS
C REPETITION FILTER, OUTPUT OF FILTER IS AT 120 KC
      DO 150 I=1,NF
C SHIFT ALL TERMS BY NRF
      IR=NRF+NF-I
      IF=NF+1-I
      XI2(IR)=XI2(IF)
150   XR2(IR)=XR2(IF)
      NRF1=NRF-1
      DO 160 I=1,NRF1
      XR2(I)=0.0
160   XI2(I)=0.0
C CALCULATE 120 KC OUTPUT
      J=0
      DO 170 L1=1,NF
```

```
      DC 17C K=1,8
      J=J+1
      DO 17C I=1,11
      LT=L1-I+92
      LK=8*(I-1)+K
      XI3(J)=XI3(J)+WREP(LK)*XI2(LT)
170   XR3(J)=XR3(J)+WREP(LK)*XR2(LT)
C J IS THE NUMBER OF OUTPUT SAMPLES FROM REP FILTER
      K2=J
      DC 18C I=1,K2
      CALL RND(XI3(I),A3,L)
180   CALL RND(XR3(I),A3,L)
      PRINT 11
11    FORMAT(1HC,4HWREP )
      CALL FTND(XR3,XI3,10,K2,120000.,FT)
C PLT SIGNAL THROUGH INTERPOLATION FILTER AND SIMULATED
C ANALOG FILTER
      DC 19C I=1,K2
      DELI=(XI3(I+1)-XI3(I))/32.0
      DELR=(XR3(I+1)-XR3(I))/32.0
      DC 19C J=1,32
      FJ=J
      K=K+1
      K1=MOD(K,100)+1
C K IS XR4 INDEX
C K1 IS XR4 CYLIC INDEX
      XI4(K1)=XI3(I)+(FJ-1.)*DELI
      XR4(K1)=XR3(I)+(FJ-1.)*DELR
      CALL RND(XR4(K1),A4,L)
      CALL RND(XI4(K1 ,A4,L)
      IF(MOD(K,16).EQ.0) GC TO 200
190   CONTINUE
      GC TO 220
200   CONTINUE
C CALCULATE OUTPUT CF ANALCG FILTER
C M IS XR5 INDEX
      M=M+1
      DO 21C L1=1,NA
      N=MCD(74+(M-1)*16+L1,100)+1
      XI5(M)=XI5(M)+ANLOG(L1)*XI4(N)
210   XR5(M)=XR5(M)+ANLOG(L1)*XR4(N)
      GO TO 190
220   CONTINUE
      M1=M/2+1
      M2=M/5+1
      DO 23C I=1,M1
      XI7(I)=XI5(2*I-1)
      XR7(I)=XR5(2*I-1)
230   CONTINUE
      PRINT 12
12    FORMAT(1HC,15HRECEIVER INPUT   )
      CALL FTND(XR7,XI7,10,M1,120000.,FT)
      DO 24C I=1,M2
      XI6(I)=XI5(5*I-4)
240   XR6(I)=XR5(5*I-4)
      PRINT 13
13    FORMAT(1H0,23HOUTPUT OF ANALOG FILTER    )
```

```
      CALL FTND(XR6,XI6,9,M2,48000.,FT)
C XR6 IS INPUT TO FFT TO LOOK AT OUTPUT SPECTRUM OF TRANSMITTER
C XR7 IS INPUT TO RECEIVER
C ADD NOISE AND SHIFT BY NRF-8
      FM1=2*M1
      DO 249 I=1,M1
249   XS=XS+XR7(I)*XR7(I)+XI7(I)*XI7(I)
      SGN=SQRT(XS/FM1)
      DO 250 I=1,M1
      MT=M1+NRF-8-I
      MI=M1+1-I
      CALL GAUSS(Z,SGM,SGN)
      ZT=ZT+Z*Z
      XR7(MT)=XR7(MI)+Z
      CALL GAUSS(Z,SGM,SGN)
      ZT=ZT+Z*Z
      XI7(MT)=XI7(MI)+Z
250   CONTINUE
      IF(ZT.EQ.0.) GO TO 251
      SG=10.0*ALOG10(XS/ZT)
251   CONTINUE
      PRINT 14
14    FORMAT(1H0,12HAFTER NOISE  )
      CALL FTND(XR7,XI7,10,M1,120000.,FT)
C SIGNAL GOES INTO REPETITION FILTER IN STEPS OF 8 INPUT SAMPLES
C OUTPUT IS AT 15 KC
      N8=NRF-9
      DO 260 I=1,N8
      XR7(I)=0.0
260   XI7(I)=0.0
C CALCULATE OUTPUT OF RECEIVER REPETITION FILTER
      M3=M1/8+1
      DO 280 K=1,M3
      DO 270 L1=1,NRF
      LT=L1+8*(K-1)
      XI8(K)=XI8(K)+WREP(L1)*XI7(LT)
 270  XR8(K)=XR8(K)+WREP(L1)*XR7(LT)
      CALL RND(XR8(K),A8,L)
280   CALL RND(XI8(K),A8,L)
      PRINT 23
23    FORMAT(1H0,4HWREP )
      CALL FTND(XR8,XI8,9,M3,15000.,FT)
C SSB 1.35KC LP FILTER
      DO 281 I=1,M3
      IS=NSB+M3-I
      IM=M3+1-I
      XR8(IS)=XR8(IM)
 281  XI8(IS)=XI8(IM)
      DO 282 I=1,NS1
      XR8(I)=0.0
282   XI8(I)=0.0
      M4=M3+NSB-1
      DO 283 I=1,M4
      DO 284 J=1,NSB
      IJ=I+J-1
      XR9(I)=XR9(I)+WSSB(J)*XR8(IJ)
 284  XI9(I)=XI9(I)+WSSB(J)*XI8(IJ)
```

```
      CALL RND(XR9(I),A9,L)
283   CALL RND(XI9(I),A9,L)
16    FORMAT(1H0,7HSSB FIL  )
      CALL FTNP(XR9,XI9,9,M4,15000.,FT)
      CALL ERR(XR9,XI9,9,M4,F1,FT)
C SHIFT BY 1.65 KC AND ADD TO GET OUTPUT
      DO 290 I=1,M4
      FI=I-1
      XW=2.0*PI*W*FI
      XR9(I)=XR9(I)+CS(XW,LF)
      XI9(I)=XI9(I)+SN(XW,LF)
      CALL RND(XR9(I),A9,L)
      CALL RND(XI9(I),A9,L)
290   XOUT(I)=XR9(I)+XI9(I)
      PRINT 20
20    FORMAT(1H0,17HOUTPUT WAVE FORM    )
      PRINT 21,(XOUT(I),I=1,M4)
21    FORMAT(1H ,10F10.3)
C
      DO 342 I=1,300
342   XI(I)=0.0
C RECEIVER OUTPUT
      DO 360 I=1,512
      FT(2,I)=CMPLX(0.,0.)
360   FT(1,I)=CMPLX(0.,0.)
      DO 370 I=1,M4
370   FT(1,I)=CMPLX(XOUT(I),0.)
      CALL FFT(FT,9,-1.)
      DO 380 I=1,512
      F2(I)=CABS(FT(2,I))
      IF(F2(I).LE.0.00001) F2(I)=0.00001
380   FTM(I)=20.0*ALOG10(F2(I))
      B=FTM(50)
      DO 381 I=1,512
381   FTM(I)=FTM(I)-B
      SMPC=15000./512.
      PRINT 385
385   FORMAT(1H1,22HRECEIVER OUTPUT IN DB    )
      PRINT 390,SMPC
390   FORMAT(1H ,16HSAMPLE SPACING =   ,F10.4)
      DO 400 I=1,52
      ~~J=I-1~~
      FJ=J
      CI=FJ*5.0*SMPC
      K1=5*J+1
      K2=K1+4
400   PRINT 350,CI,(FTM(K3),K3=K1,K2)
~~350   FORMAT(1H ,6F12.2)~~
C
      ER=0.0
      READ 435,(FM(I),I=1,512)
      DO 410 I=1,512
      ER2=ER2+FM(I)*FM(I)
~~410   ER=ER+(F1(I)-FM(I))**2~~
      PRINT 420,ER
420   FORMAT (1H0,24HINTEGRAL SQUARED NOISE = ,F8.3)
      PRINT 425,SG
```

```
 425  FCRMAT(1HC,44HSIGNAL TO NCISE RATIC AT INPUT TO RECEIVER =  ,F9.2,
     1  4H DB  )
      SGA=10.0*ALCG10(ER2/ER)
      PRINT 430,SGA
430   FCRMAT(1HC,27HSIGNAL TO NCISE PCWER CUT =   ,F10.3,3H DB )
      PUNCH 435,(F1(I),I=1,512)
435   FCRMAT(10F8.5)
999   CONTINUE
      STOP
      END




      SUBROUTINE GAUSS(X,SCH,SGN)
C THIS SUBROUTINE RETURNS A GAUSSIAN RANDOM VARIABLE X
C WITH ZERO MEAN AND ST. DEV. = SGH/(10.**(SGN/20.))
C SGH = SQUARE ROOT OF THE SUM OF THE SQUARES OF THE H"S
C SGN = VOLTAGE SIGNAL TO NOISE RATIO IN DB
      DIMENSION INTG(12),IND(12),IP(12),IMX(12)
      K=K+1
      IF(K.GT.1) GO TO 20
      SD=SGH/(10.**(SGN/20.))
      INTG(1)=91546128
      DO 10 I=1,11
      INTG(I+1)=INTG(I)*101+331
      IND(I)=INTG(I+1)/100000000
      IND(I)=IND(I)*100000000
10    INTG(I+1)=INTG(I+1)-IND(I)
      DO 15 I=1,12
  15  IP(I)=(I+9)**2-(I+9)+41
20    CONTINUE
      SUM=0.0
      DO 30 I=1,12
      INTG(I)=INTG(I)*101+IP(I)
      IND(I)=INTG(I)/100000000
      IND(I)=IND(I)*100000000
      INTG(I)=INTG(I)-IND(I)
      IMX(I)=INTG(I)/1000000
30    SUM=SUM+FLOAT(IMX(I))
      FN=(SUM-594.0)/99.995
C FN IS A GAUSSIAN RV WITH ZERO MEAN AND VARIANCE = 1.0
      X=FN*SD
      RETURN
      END
```

```
      SUBROUTINE RND(X,XMAX,L)
C THIS SUBROUTINE ROUNDS X TO L BIT PRECISION
C XMAX IS THE MAXIMUM MAGNITUDE OF THE VARIABLE X
      S=2**(L-1)-1
      IF(X.GE.0.) IX=X*(S/XMAX)+.5
      IF(X.LT.0.) IX=X*(S/XMAX)-.5
      FIX=IX
      X=FIX*(XMAX/S)
      RETURN
      END
```

```
      FUNCTION CS(X,L)
      CS=COS(X)
      CALL RND(CS,1.0,L)
      RETURN
      END
```

```
      FUNCTION SN(X,L)
      SN=SIN(X)
      CALL RND(SN,1.0,L)
      RETURN
      END
```

```
      SUBROUTINE FFT(X,NSTAGE,SIGN)
C FAST FOURIER TRANSFORM SUBROUTINE
C COMMON VARIABLES
C X(2,1024)= DATA'   COMPLEX. INPUT IN COLUMN 1, OUTPUT FOR
C INVERSE TRANSFORM IN COLUMN 1. NORMALIZED OUTPUT FOR
C FORWARD TRANSFORM IN COLUMN 2. UNNORMALIZED FORWARD TRANSFORM
C OUTPUT IN COLUMN 1. NSTAGE= NUMBER OF STAGES AND POWER
C OF TWO WHICH N IS; N=2**NSTAGE
C SIGN= IDENTIFIES DIRECTION OF TRANSFORM
C SIGN=-1.; FORWARD TRANSFORM.
C SIGN=+1.; INVERSE TRANSFORM.
      COMPLEX X(2,1024),W
      INTEGER R
      N=2**NSTAGE
      N2=N/2
      FLTN=N
      PHI2N=6.2831853/FLTN
      DO 3 J=1,NSTAGE
      N2J=N/(2**J)
      NR=N2J
      NI=(2**J)/2
      DO 2 I=1,NI
      IN2J=(I-1)*N2J
      FLIN2J=IN2J
      TEMP=FLIN2J*PHI2N*SIGN
      W=CMPLX(COS(TEMP),SIN(TEMP))
      DO 2 R=1,NR
      ISUB=R+IN2J
      ISUB1=R+IN2J*2
      ISUB2=ISUB1+N2J
      ISUB3=ISUB+N2
      X(2,ISUB)=X(1,ISUB1)+W*X(1,ISUB2)
      X(2,ISUB3)=X(1,ISUB1)-W*X(1,ISUB2)
2     CONTINUE
      DO 3 R=1,N
3     X(1,R)=X(2,R)
      IF(SIGN.GT.0.) RETURN
      DO 4 R=1,N
4     X(2,R)=X(1,R)/FLTN
      RETURN
      END
```

```
      SUBROUTINE ERR(XR,XI,IL,M,FM,FT)
      DIMENSION FM(512),XR(512),XI(512)
      COMPLEX FT(2,512)
      DO 10 I=1,512
      FM(I)=0.0
      FT(1,I)=CMPLX(0.,0.)
10    FT(2,I)=CMPLX(0.,0.)
      N=2**IL
      DO 20 I=1,M
20    FT(1,I)=CMPLX(XR(I),XI(I))
      CALL FFT(FT,IL,-1.)
      DO 30 I=1,N
30    FM(I)=CABS(FT(2,I))
      A=FM(1)
      DO 40 I=1,N
 40   FM(I)=FM(I)/A
      RETURN
      END
```

INPUT WAVE FORM
-0.01587 0.01587 -0.09524 -0.04762 0.55556 1.00000 0.55556 -0.04762 -0.09524 0.01587
0.01587

WSB

SAMPLE SPACING = 29.2969

Spectrum in DB

0. FREQ.					
0.	0.	-0.00	0.00	0.02	0.03
146.48	0.05	0.05	0.05	0.04	0.02
292.97	0.00	-0.01	-0.02	-0.02	-0.00
439.45	0.01	0.03	0.05	0.05	0.05
585.94	0.03	0.01	-0.02	-0.04	-0.05
732.42	-0.05	-0.04	-0.02	0.01	0.03
878.91	-0.04	0.04	0.04	0.03	0.03
1025.39	0.04	0.07	0.11	0.16	0.19
1171.87	0.18	0.10	-0.09	-0.41	-0.90
1318.36	-1.59	-2.51	-3.69	-5.18	-7.01
1464.84	-9.22	-11.89	-15.11	-18.99	-23.77
1611.33	-29.80	-37.95	-51.36	-62.11	-61.49
1757.81	-65.64	-64.32	-65.64	-54.51	-47.54
1904.30	-44.04	-42.53	-42.61	-44.32	-48.23
2050.78	-57.09	-63.55	-55.22	-56.61	-65.64
2197.27	-51.12	-44.48	-40.83	-38.75	-37.76
2343.75	-37.66	-38.35	-39.79	-41.97	-44.83
2490.23	-48.15	-51.22	-52.78	-52.19	-50.41
2636.72	-48.63	-47.39	-46.86	-47.06	-47.99
2783.20	-49.68	-52.19	-55.64	-60.37	-65.64
2929.69	-65.64	-65.64	-65.64	-64.84	-62.31
3076.17	-60.04	-57.94	-55.96	-54.00	-52.00
3222.66	-49.94	-47.87	-45.90	-44.15	-42.73
3369.14	-41.74	-41.30	-41.54	-42.71	-45.36
3515.62	-54.41	-64.44	-47.74	-42.53	-39.73
3662.11	-38.17	-37.47	-37.45	-38.00	-39.03
3808.59	-40.45	-42.11	-43.82	-45.40	-46.78
3955.08	-48.19	-50.23	-54.42	-65.64	-53.82
4101.56	-16.56	-42.49	-39.90	-38.32	-37.57
4248.05	-37.58	-38.39	-40.14	-43.21	-48.79
4394.53	-65.64	-52.28	-46.68	-44.31	-43.32
4541.02	-43.15	-43.51	-44.23	-45.19	-46.34
4687.50	-47.76	-49.47	-52.59	-57.95	-65.64
4833.98	-59.05	-53.30	-50.02	-50.06	-50.83
4980.47	-53.49	-60.01	-65.64	-57.20	-53.36
5126.95	-52.44	-53.72	-58.04	-65.64	-60.92
5273.44	-55.50	-54.24	-56.04	-64.06	-62.79
5419.92	-53.20	-49.29	-47.63	-47.61	-49.32
5566.41	-53.76	-65.64	-57.09	-51.20	-49.27
5712.89	-49.67	-52.77	-63.87	-57.13	-49.27
5859.37	-45.76	-44.04	-43.49	-43.86	-45.02
6005.86	-46.83	-49.07	-51.15	-52.16	-51.53
6152.34	-48.77	-47.77	-46.10	-44.58	-44.47
6298.83	-44.60	-45.37	-46.83	-49.12	-52.62
6445.31	-58.74	-65.64	-58.40	-52.19	-48.32
6591.80	-45.42	-43.12	-41.32	-39.99	-39.15
6738.28	-38.83	-39.08	-39.99	-41.72	-44.59
6884.77	-49.46	-60.46	-59.57	-52.40	-50.53
7031.25	-51.05	-53.64	-61.26	-65.44	-54.89
7177.73	-53.68	-53.28	-55.42	-62.84	-64.28
7324.22	-53.46	-48.79	-46.08	-44.37	-43.41
7470.70	-43.01	-43.11	-43.70	-44.85	-46.72
7617.19	-49.78	-54.79	-65.64	-61.92	-55.48
7763.67	-54.06	-55.93	-65.64	-57.93	-49.06

7910.16	-44.43	-41.54	-39.67	-38.65	-38.32
8056.64	-38.65	-39.63	-41.29	-43.68	-46.85
8203.12	-50.75	-54.74	-56.90	-55.58	-52.37
8349.61	-49.20	-46.67	-44.80	-43.50	-42.62
8496.09	-42.04	-41.66	-41.39	-41.21	-41.12
8642.58	-41.20	-41.54	-42.29	-43.63	-45.81
8789.06	-49.36	-55.76	-65.64	-59.35	-56.12
8935.55	-57.67	-65.64	-58.84	-50.75	-46.74
9082.03	-44.52	-43.47	-43.36	-44.06	-45.48
9228.52	-47.47	-49.63	-51.13	-51.10	-49.59
9375.00	-47.54	-45.77	-44.69	-44.53	-45.56
9521.48	-48.45	-56.06	-59.44	-48.47	-44.23
9667.97	-42.12	-41.28	-41.43	-42.50	-44.54
9814.45	-47.72	-52.43	-59.57	-65.64	-65.64
9960.94	-65.64	-59.02	-54.06	-50.97	-49.01
10107.42	-47.76	-46.92	-46.23	-45.52	-44.69
10253.91	-43.79	-42.93	-42.25	-41.88	-41.92
10400.39	-42.44	-43.52	-45.20	-47.49	-50.24
10546.87	-52.94	-54.50	-54.08	-52.31	-50.41
10693.36	-49.14	-48.87	-49.99	-53.44	-64.65
10839.84	-58.13	-50.37	-47.02	-45.63	-45.70
10986.33	-47.26	-50.95	-60.07	-61.73	-53.03
11132.81	-50.59	-50.90	-54.23	-65.64	-59.09
11279.30	-50.00	-46.87	-45.57	-45.59	-46.86
11425.78	-49.65	-55.11	-65.64	-59.01	-53.62
11572.27	-51.57	-50.90	-50.90	-51.03	-50.83
11718.75	-50.03	-48.71	-47.16	-45.68	-44.46
11865.23	-43.58	-43.09	-42.99	-43.31	-44.06
12011.72	-45.33	-47.31	-50.49	-53.49	-65.64
12158.20	-55.35	-49.91	-47.15	-45.83	-45.67
12304.69	-46.80	-49.82	-57.68	-60.59	-50.04
12451.17	-46.10	-44.38	-44.18	-45.41	-48.53
12597.66	-55.55	-65.64	-53.56	-50.45	-50.53
12744.14	-53.59	-63.72	-60.99	-53.79	-52.40
12890.62	-55.09	-65.64	-54.48	-48.05	-45.41
13037.11	-45.07	-47.21	-54.67	-57.09	-46.89
13183.59	-43.64	-43.36	-46.13	-56.30	-54.10
13330.08	-49.83	-61.65	-37.38	-28.40	-22.30
13476.56	-17.63	-13.98	-11.00	-8.56	-6.57
13623.05	-4.97	-3.70	-2.71	-1.96	-1.43
13769.53	-1.07	-0.85	-0.73	-0.68	-0.66
13916.02	-0.64	-0.62	-0.58	-0.52	-0.44
14062.50	-0.37	-0.30	-0.24	-0.20	-0.18
14208.98	-0.17	-0.16	-0.15	-0.13	-0.11
14355.47	-0.08	-0.05	-0.02	0.01	0.02
14501.95	0.03	0.02	0.02	0.01	0.02
14648.44	0.03	0.05	0.08	0.11	0.13
14794.92	0.15	0.15	0.13	0.11	0.08

WREP

SAMPLE SPACING = 117.1875

0.	0.	0.02	0.03	-0.01	0.02
585.94	-0.03	-0.12	-0.03	0.03	0.04
1171.87	0.08	-1.04	-5.29	-15.00	-37.05
1757.81	-46.61	-38.37	-39.83	-46.89	-52.94
2343.75	-35.99	-37.84	-43.15	-47.08	-60.92
2929.69	-46.22	-47.53	-53.69	-45.22	-37.21
3515.62	-40.69	-40.93	-36.05	-40.15	-44.89
4101.56	-46.32	-34.98	-39.36	-44.83	-41.36

RECEIVER OUTPUT IN DB
SAMPLE SPACING = 29.2969

0.	-59.18	-58.98	-52.51	-42.49	-33.70
146.48	-26.39	-20.36	-15.42	-11.40	-8.16
292.97	-5.62	-3.66	-2.22	-1.22	-0.58
439.45	-0.22	-0.06	-0.02	-0.05	-0.08
585.94	-0.09	-0.08	-0.05	-0.00	0.05
732.42	0.08	0.10	0.11	0.11	0.10
878.91	0.08	0.06	0.03	-0.01	-0.06
1025.39	-0.11	-0.15	-0.16	-0.14	-0.09
1171.87	-0.01	0.08	0.16	0.21	0.22
1318.36	0.20	0.15	0.09	0.04	0.
1464.84	-0.01	0.02	0.07	0.12	0.18
1611.33	0.23	0.27	0.30	0.32	0.33
1757.81	0.33	0.31	0.28	0.23	0.16
1904.30	0.08	0.01	-0.03	-0.03	0.02
2050.78	0.10	0.20	0.28	0.32	0.29
2197.27	0.21	0.06	-0.10	-0.25	-0.35
2343.75	-0.37	-0.32	-0.22	-0.10	-0.00
2490.23	0.04	0.01	-0.09	-0.25	-0.42
2636.72	-0.58	-0.69	-0.73	-0.72	-0.68
2783.20	-0.67	-0.75	-0.96	-1.37	-2.03
2929.69	-2.99	-4.30	-6.04	-8.27	-11.09
3076.17	-14.60	-18.96	-24.37	-31.18	-40.01
3222.66	-52.08	-62.10	-62.55	-62.12	-62.61
3369.14	-63.73	-63.17	-63.38	-66.41	-71.90
3515.62	-70.59	-65.88	-62.43	-60.85	-60.97
3662.11	-61.95	-62.83	-62.34	-59.57	-57.28
3808.59	-57.11	-59.59	-65.28	-68.60	-66.48
3955.08	-64.47	-61.44	-59.93	-60.67	-62.93
4101.56	-64.37	-64.93	-66.54	-67.95	-67.87
4248.05	-68.94	-70.77	-67.32	-63.53	-62.06
4394.53	-62.81	-65.58	-69.36	-73.16	-69.13
4541.02	-62.44	-59.16	-58.49	-59.97	-63.17
4687.50	-67.85	-71.72	-64.62	-60.17	-58.69
4833.98	-59.37	-61.53	-63.76	-64.04	-62.71
4980.47	-62.04	-63.64	-68.98	-69.01	-63.89
5126.95	-62.36	-62.69	-63.32	-63.96	-65.41
5273.44	-67.10	-67.47	-67.92	-70.05	-73.12
5419.92	-74.71	-76.25	-78.72	-78.66	-76.34
5566.41	-72.68	-69.99	-70.01	-72.65	-68.78
5712.89	-63.51	-61.10	-60.99	-62.94	-66.81
5859.37	-70.40	-68.50	-65.46	-63.52	-62.38
~~6005.86~~	~~-61.52~~	-61.09	-61.99	-65.40	-74.35
6152.34	-76.47	-72.01	-75.83	-92.16	-76.20
6298.83	-76.38	-83.07	-78.45	-76.59	-74.29
6445.31	-69.38	-67.17	-67.33	-66.98	-64.80
6591.80	-63.82	-65.34	-69.97	-75.46	-73.97
6738.28	-73.80	[illegible]	-69.23	-68.36	-67.80
~~6884.77~~	~~-65.55~~	-63.46	-63.45	-64.87	-67.19
7031.25	-70.01	-76.90	-78.45	-68.50	-66.02
7177.73	-67.71	-76.77	-75.96	-69.71	-71.83
7324.22	-90.89	-68.70	-64.47	-64.22	-66.94
7470.70	-72.43	-77.24	-72.43	-66.94	-64.22

~~INTEGRAL SQUARED NOISE = 6.67~~

SIGNAL TO NOISE RATIO AT INPUT TO RECEIVER = 14.96 DB

SIGNAL TO NOISE POWER OUT = 30.807 DB

3. DSB SIMULATION

```
C DOUBLE SIDEBAND SIMULATION   D.L.FLETCHER
      DIMENSION WSSB(100),WREP(100),ANLOG(100),X(100),XR(300),
     *XI(300),XR2(400),XI2(400),XR3(850),XI3(350),XR4(150),XI4(150),
     *XR5(1650),XI5(1650),XR6(350),XI6(350),XR7(900),XI7(900),
     *XR8(300),XI8(300),XR9(300),XI9(300),XOUT(300),FTM(1030)
      DIMENSION FM(512),F1(512)
      COMPLEX FT(2,1024),C
      DATA A1,A3,A4,A8,A9/1.284,2.63,2.63,14.32,19.02/
      DATA LF,L/10,8/
      DATA MSIG,MS,MR,MA/5,45,40,20/
      DATA NS,NSB,NRF,NA/11,91,81,41/
C NS=NO. OF TERMS IN INPUT SIGNAL = 2*MSIG+1
C NSB=NO. OF TERMS IN DSB FILTER = 2*MS+1
C NRF=NO. OF TERMS IN REP FILTER = 2*MR+1
C NA=NO. OF TERMS IN ANALOG FILTER = 2*MA+1
      DATA FZS,FZR,FZA/.4266,.1258,.0312/
      DATA SGN/12./
      PI=3.14159265
C CALCULATE TAP WEIGHTS
      DO 50 I=1,MS
      FI =I
      WIN=.54+.46*COS(PI*FI/48.)
      IMS=MS+1+I
      WSSB(IMS)=WIN*SIN(FZS*PI*FI)/(PI*FI*FZS)
      CALL RND(WSSB(IMS),1.0,LF)
      JMS=MS+1-I
 50   WSSB(JMS)=WSSB(IMS)
      WSSB(MS+1)=1.0
      CALL RND(WSSB(MS+1),1.0,LF)
      FMR=MR
      DO 60 I=1,MR
      FI=I
      WIN=.54+.46*COS(PI*FI/FMR)
      IMR=MR+1+I
      WREP(IMR)=WIN*SIN(FZR*PI*FI)/(PI*FI*FZR)
      CALL RND(WREP(IMR),1.0,LF)
      JMR=MR+1-I
 60   WREP(JMR)=WREP(IMR)
      WREP(MR+1)=1.0
      CALL RND(WREP(MR+1),1.0,LF)
      FMA=MA
      DO 70 I=1,MA
      FI=I
      WIN=.54+.46*COS(PI*FI/FMA)
      IMA=MA+1+I
      ANLOG(IMA)=WIN*SIN(FZA*PI*FI)/(PI*FI*FZA)
      JMA=MA+1-I
      ANLOG(JMA)=ANLOG(IMA)
 70   CONTINUE
      ANLOG(MA+1)=1.0
CLOAD 50% RC PULSE! BW=4KC
      A=8.0*PI/15.0
      DO 80 I=1,MSIG
      IMSIG=MSIG+1+I
      X(IMSIG)=SIN(A*FI)*COS(5*A*FI)/(A*FI*(1.-(8.*FI/15.)**2))
      CALL RND(X(IMSIG),1.0,7)
```

```
      JMSIG=MSIG+1-I
80    X(JMSIG)=X(IMSIG)
      X(MSIG+1)=1.0
      CALL RND(X(MSIG+1),1.0,7)
      DO 110 I=1,NS
110   XR(I)=X(I)
C CONVOLVE WITH FIRST FILTER
      DO 112 I=1,NS
      IN=NSB+NS-I
      IS=NS+1-I
C SHIFT ALL TERMS BY NSB
      XR(IN)=XR(IS)
112   XI(I)=0.0
      NS1=NSB-1
      DO 113 I=1,NS1
      XR(I)=0.0
113   XI(I)=0.0
C NF=NO. OF TERMS OUT OF FIRST FILTER
      NF=NSB+NS-1
      DO 115 I=1,NF
      DO 114 J=1,NSB
      IJ=I+J-1
      XR2(I)=XR2(I)+WSSB(J)*XR(IJ)
114   XI2(I)=0.0
115   CALL RND(XR2(I),A1,L)
      PRINT 10
10    FORMAT(1H0,30HWAVE FORM OUT OF FIRST FILTER   )
      PRINT 11,(XR2(I),I=1,NF)
11    FORMAT(1H ,10F12.3)
CEFFECTIVELY FILL WITH ZEROS AND CONVOLVE WITH 7 KC LOW-PASS
C REPETITION FILTER, OUTPUT OF FILTER IS AT 120 KC
      DO 150 I=1,NF
      IN=NRF+NF-I
      IF=NF+1-I
C SHIFT ALL TERMS BY
150   XR2(IN)=XR2(IF)
      NRF1=NRF-1
      DO 160 I=1,NRF1
      XR2(I)=0.0
160   XI2(I)=0.0
C CALCULATE 120 KC OUTPUT
      J=0
      DO 170 L1=1,NF
      DO 170 K=1,8
      J=J+1
      DO 170 I=1,11
      LT=L1-I+92
      LK=8*(I-1)+K
170   XR3(J)=XR3(J)+WREP(LK)*XR2(LT)
C J IS THE NUMBER OF OUTPUT SAMPLES FROM REP FILTER
      K2=J
      DO 180 I=1,K2
180   CALL RND(XR3(I),A3,L)
C INTERPOLATION AND ANALOG FILTER
      DO 190 I=1,K2
      DELR=(XR3(I+1)-XR3(I))/32.0
      DO 190 J=1,32
```

```
      K=K+1
      K1=MOD(K,100)+1
C K IS XR4 INDEX
C K1 IS XR4 CYLIC INDEX
      FJ=J
      XR4(K1)=XR3(I)+(FJ-1.)*DELR
      CALL RND(XR4(K1),A4,L)
      IF(MOD(K,16).EQ.0) GO TO 200
190   CONTINUE
      GO TO 220
200   CONTINUE
C CALCULATE OUTPUT OF ANALOG FILTER
C M IS XR5 INDEX
      M=M+1
      DO 210 L1=1,NA
      N=MOD(74+(M-1)*16+L1,100)+1
210   XR5(M)=XR5(M)+ANLOG(L1)*XR4(N)
      GO TO 190
220   CONTINUE
      M1=M/2+1
      M2=M/5+1
      DO 230 I=1,M1
      XI7(I)=0.0
      XR7(I)=XR5(2*I-1)
230   CONTINUE
      PRINT 13
13    FORMAT(1H0,3HXR7 )
      PRINT 11,(XR7(I),I=1,M1)
      DO 240 I=1,M2
      XI6(I)=0.0
240   XR6(I)=XR5(5*I-4)
      PRINT 25
25    FORMAT(1H0, 23HOUTPUT OF ANALOG FILTER  )
      CALL FTND(XR6,XI6,9,M2,4P000.,FT)
241   CONTINUE
C XR6 IS INPUT TO FFT TO LOOK AT OUTPUT SPECTRUM OF TRANSMITTER
C XR7 IS INPUT TO RECEIVER
C ADD NOISE AND SHIFT BY NRF-8
      FM1=M1
      DO 249 I=1,M1
249   XS=XS+XR7(I)*XR7(I)
      SQH=SQRT(XS/FM1)
      DO 250 I=1,M1
      CALL GAUSS(Z,SQH,SGV)
      ZT=ZT+Z*Z
      IN=M1+NRF-8*I
      IM=M1+1-I
      XR7(IN)=XR7(IM)+Z
250   CONTINUE
      PRINT 11,(XR7(I),I=1,M1)
      IF(ZT.EQ.0.) GO TO 251
      SG=10.0*ALOG10(XS/ZT)
251   CONTINUE
C SIGNAL GOES INTO REPETITION FILTER IN STEPS OF 8 INPUT SAMPLES
C OUTPUT IS AT 1.5 KC
      N8=NRF-9
      DO 260 I=1,N8
```

```
      XR7(I)=0.0
260   XI7(I)=0.0
C CALCULATE OUTPUT OF RECEIVER REPETITION FILTER
      M3=M1/8+1
      DO 280 K=1,M3
      DO 270 L1=1,NRF
      LT=L1+8*(K-1)
270   XR8(K)=XR8(K)+WREP(L1)*XR7(LT)
280   CALL RND(XR8(K),A8,L)
C DSB 3KC LP FILTER
      DO 281 I=1,M3
      ISB=NSB+M3-I
      IM=M3+1-I
      XR8(ISB)=XR8(IM)
281   XI8(I)=0.0
      DO 282 I=1,NS1
      XR8(I)=0.0
282   XI8(I)=0.0
      M4=M3+NSB-1
      DO 283 I=1,M4
      XR9(I)=0.0
      DO 284 J=1,NS3
      IJ=I+J-1
      XR9(I)=XR9(I)+WSSB(J)*XR8(IJ)
284   XI9(I)=0.0
      CALL RND(XR9(I),A9,L)
283   CONTINUE
      DO 290 I=1,M4
290   XOUT(I)=XR9(I)
C RECEIVER OUTPUT
350   FORMAT(1H ,6F12.2)
      DO 360 I=1,300
360   XI9(I)=0.0
      PRINT 385
385   FORMAT(1H0,22HRECEIVER OUTPUT IN DB   )
      CALL FIND(XOUT,XI9,9,M4,15000.,FT)
      CALL FPR(XOUT,0,M4,FM)
C
      FR=0.0
      READ 435,(F1(I),I=1,512)
      DO 410 I=1,512
      FR2=FR2+F1(I)*F1(I)
410   FR=FR+(F1(I)-FM(I))**2
      PRINT 420,FR
420   FORMAT(1H0,24HINTEGRAL SQUARED NOISE =  ,F10.3)
      PRINT 425,SG
425   FORMAT(1H0,23HSIGNAL TO NOISE RATIO =  ,F10.2,3H DB   )
      SGA=10.0*ALOG10(FR2/FR)
      PRINT 430,SGA
430   FORMAT(1H0,27HSIGNAL TO NOISE POWER OUT =  ,F10.3,3H DB )
435   FORMAT(10F8.5)
999   CONTINUE
      STOP
      END
```

```
      SUBROUTINE FIND(XR,XI,IL,M,REP,FT)
C XR,XI ELEMENTS OF  COMPLEX DATA SEQUENCE
C IL - SIZE OF TRANSFORM IS 2**IL
C M - NUMBER OF ELEMENTS IN INPUT DATA SEQUENCE
C REP - REPETITION RANGE OF TRANSFORM
      DIMENSION FM(1030),XR(1024),XI(1024)
      COMPLEX FT(2,1024)
      DO 10 I=1,1024
      FM(I)=0.0
      FT(1,I)=CMPLX(0.,0.)
10    FT(2,I)=CMPLX(0.0,0.0)
      N=2**IL
      FN=N
      SMPC=REP/FN
      DO 20 I=1,M
20    FT(1,I)=CMPLX(XR(I),XI(I))
      CALL FFT(FT,IL,-1.)
      DO 30 I=1,N
      B=CABS(FT(2,I))
      IF(B.LE.0.00001) B=0.00001
30    FM(I)=20.0*ALOG10(B)
      A=FM(1)
      DO 35 I=1,N
35    FM(I)=FM(I)-A
      PRINT 40,SMPC
40    FORMAT(1H0,14HSAMPLE SPACING = ,F10.4)
      NT=N/5+1
      DO 50 I=1,NT
      J=I-1
      FJ=J
      CI=FJ*5.0*SMPC
      K1=5*J +1
      K2=K1+4
50    PRINT 60,CI,(FM(K3),K3=K1,K2)
60    FORMAT(1H ,6F12.2)
      RETURN
      END
```

```
      SUBROUTINE ERR(XR,TL,M,FM)
      DIMENSION FM(1030),XR(1024)
      COMPLEX FT(2,512)
      DO 10 I=1,512
      FM(I)=0.0
      FT(1,I)=CMPLX(0.,0.)
10    FT(2,I)=CMPLX(0.,0.)
      N=2**I
      DO 20 I=1,M
20    FT(1,I)=CMPLX(XR(I),0.)
      CALL FFT(FT,TL,-1.)
      DO 30 I=1,N
30    FM(I)=CABS(FT(2,I))
      A=FM(1)
      DO 40 I=1,N
40    FM(I)=FM(I)/A
      RETURN
      END
```

OUTPUT OF ANALOG FILTER

SAMPLE SPACING = 93.7500

0.	0.	0.02	0.06	0.05	0.02
468.75	0.03	0.08	0.07	-0.03	-0.10
937.50	0.00	0.22	0.31	0.16	-0.13
1406.25	-0.26	-0.12	0.08	0.08	-0.09
1875.00	-0.18	-0.03	0.20	0.24	0.08
2343.75	-0.09	-0.09	0.00	0.04	0.06
2812.50	0.19	0.31	-0.09	-1.70	-5.26
3281.25	-11.87	-25.14	-34.81	-37.82	-46.18
3750.00	-45.59	-48.66	-40.96	-41.45	-40.39
4218.75	-30.74	-35.21	-45.92	-31.47	-37.00
4687.50	-35.19	-33.43	-45.07	-31.06	-29.68
5156.25	-44.72	-33.82	-35.84	-40.30	-32.12
5625.00	-37.61	-41.15	-36.05	-47.91	-40.54
6093.75	-41.18	-40.35	-39.48	-38.99	-39.65
6562.50	-38.27	-30.07	-43.45	-53.40	-42.58
7031.25	-41.03	-30.07	-36.48	-37.89	-58.13
7500.00	-35.23	-35.76	-38.16	-46.82	-37.00
7968.75	-37.21	-30.47	-41.54	-51.27	-40.82
8437.50	-34.65	-36.25	-49.55	-49.37	-42.54
8906.25	-32.02	-33.55	-54.19	-53.05	-33.36
9375.00	-50.05	-45.41	-47.69	-32.78	-32.86
9843.75	-54.56	-31.08	-30.01	-36.41	-57.21
10312.50	-47.31	-36.85	-37.06	-53.75	-44.19
10781.25	-67.31	-30.28	-38.24	-49.86	-54.82
11250.00	-44.44	-35.96	-39.52	-39.48	-31.60
11718.75	-34.70	-41.84	-30.92	-31.88	-44.37
12187.50	-41.12	-45.73	-47.53	-39.35	-42.14
12656.25	-53.73	-61.42	-58.02	-61.13	-51.01
13125.00	-44.62	-43.07	-42.62	-41.70	-42.61
13593.75	-55.39	-41.94	-36.20	-36.58	-44.95
14062.50	-46.56	-41.46	-49.59	-44.22	-38.14
14531.25	-39.36	-50.50	-47.25	-45.45	-54.38
15000.00	-41.53	-30.74	-44.80	-60.16	-46.73
15468.75	-47.71	-54.42	-65.30	-51.98	-45.07
15937.50	-42.43	-46.45	-57.89	-53.31	-63.23
16406.25	-48.00	-45.01	-43.78	-48.19	-53.02
16875.00	-57.47	-55.75	-42.88	-37.45	-35.15
17343.75	-34.40	-34.38	-35.47	-41.40	-45.84
17812.50	-35.24	-34.07	-39.66	-45.50	-35.74
18281.25	-34.86	-30.13	-47.25	-51.85	-44.47
18750.00	-46.19	-55.24	-41.69	-39.26	-42.22
19218.75	-53.57	-56.53	-53.88	-53.96	-52.99
19687.50	-56.71	-45.79	-38.19	-36.02	-38.35
20156.25	-46.87	-40.49	-48.27	-53.38	-50.47
20625.00	-45.02	-41.47	-40.71	-44.46	-49.26
21093.75	-45.31	-49.82	-44.41	-43.29	-51.65
21562.50	-55.00	-46.75	-35.68	-33.87	-39.05
22031.25	-48.33	-30.31	-40.40	-51.04	-59.34
22500.00	-56.35	-58.22	-51.47	-60.27	-39.15
22968.75	-34.42	-36.82	-55.18	-37.45	-39.92
23437.50	-47.89	-37.43	-39.79	-56.03	-59.80
23906.25	-45.78	-46.07	-45.78	-59.80	-56.03
24375.00	-39.79	-37.43	-47.89	-39.92	-37.55
24843.75	-55.18	-36.82	-34.42	-39.15	-60.27

25312.50	-51.47	-55.52	-56.35	-59.34	-51.04				
25781.25	-40.40	-38.71	-48.33	-39.05	-33.87				
26250.00	-35.68	-44.55	-55.00	-51.65	-43.29				
26718.75	-44.41	-48.82	-45.31	-49.28	-44.66				
27187.50	-40.71	-41.47	-45.02	-50.47	-53.38				
27656.25	-48.27	-40.49	-46.87	-38.35	-36.02				
28125.00	-38.19	-45.99	-56.71	-52.99	-53.96				
28593.75	-53.88	-55.53	-53.57	-42.72	-39.26				
29062.50	-41.69	-52.54	-46.19	-44.47	-51.65				
29531.25	-47.25	-38.73	-34.88	-39.74	-45.50				
30000.00	-39.86	-34.07	-35.24	-45.84	-41.40				
30468.75	-35.67	-34.38	-34.40	-39.15	-37.45				
30937.50	-42.88	-55.75	-57.47	-53.02	-48.19				
31406.25	-43.78	-45.91	-48.00	-53.23	-53.51				
31875.00	-57.89	-44.45	-42.43	-45.07	-51.98				
32343.75	-65.30	-54.82	-47.71	-46.73	-60.16				
32812.50	-44.80	-30.74	-41.53	-54.38	-43.65				
33281.25	-47.25	-50.90	-39.36	-38.14	-44.22				
33750.00	-49.59	-41.46	-46.56	-44.95	-36.68				
34218.75	-38.20	-47.94	-59.39	-49.51	-41.70				
34687.50	-42.62	-45.07	-44.62	-51.01	-61.13				
35156.25	-58.02	-61.42	-59.73	-42.14	-39.55				
35625.00	-47.53	-45.73	-41.12	-44.37	-31.88				
36093.75	-30.92	-41.84	-34.70	-31.60	-39.08				
36562.50	-39.52	-35.06	-44.44	-54.92	-49.86				
37031.25	-38.24	-39.58	-67.31	-44.19	-53.75				
37500.00	-37.06	-34.85	-47.31	-57.21	-36.41				
37968.75	-30.01	-31.08	-54.56	-32.86	-32.78				
38437.50	-47.69	-45.41	-50.05	-33.36	-33.05				
38906.25	-54.19	-33.55	-32.92	-42.94	-49.37				
39375.00	-49.55	-34.95	-34.45	-40.82	-51.27				
39843.75	-41.54	-39.47	-37.21	-37.00	-46.87				
40312.50	-38.16	-35.76	-35.23	-58.13	-37.89				
40781.25	-36.48	-30.07	-44.03	-49.58	-53.60				
41250.00	-43.45	-38.07	-38.27	-39.45	-38.99				
41718.75	-39.48	-40.35	-41.18	-40.54	-47.91				
42187.50	-36.05	-41.15	-37.61	-32.12	-40.30				
42656.25	-35.84	-33.82	-44.72	-29.48	-31.06				
43125.00	-45.07	-36.43	-35.19	-37.00	-31.67				
43593.75	-45.92	-35.21	-30.74	-40.39	-42.45				
44062.50	-40.96	-48.46	-45.59	-45.18	-37.82				
44531.25	-34.81	-25.14	-11.87	-5.26	-1.70				
45000.00	-0.09	0.31	0.19	0.06	0.04				
45468.75	0.00	-0.09	-0.09	0.08	0.24				
45937.50	0.20	-0.03	-0.18	-0.09	0.08				
46406.25	0.08	-0.12	-0.26	-0.13	0.14				
46875.00	0.31	0.22	0.00	-0.10	-0.03				
47343.75	0.07	0.08	0.03	0.02	0.05				
47812.50	0.06	0.02	0.	0.	0.				
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.047	-0.059	0.532	-0.443	-0.552	-0.163	-0.776	-0.067
1.634	0.405	0.377	-1.340	-1.151	0.407	-0.126	0.645	-0.414	0.924
0.200	-1.087	-0.429	-0.288	0.091	-0.562	1.227	-0.503	-0.355	1.390
0.229	0.007	0.405	-0.749	0.303	0.946	-0.335	0.939	0.015	-0.310

-1.330	1.057	-0.089	0.510	-0.549	0.155	-0.244	0.015	0.665	-0.473
-0.237	0.479	0.609	0.920	-0.990	-0.329	-0.747	0.429	0.111	0.739
-0.303	0.569	0.540	-0.044	0.148	-0.259	0.547	1.656	-0.549	-0.643
-1.020	1.323	-1.478	0.495	0.429	-0.495	-0.946	-1.604	-0.059	-0.370
0.443	1.072	1.183	-0.059	0.495	0.251	-1.655	0.096	0.650	-0.037
-0.488	0.392	-0.525	-0.214	-0.599	0.355	1.190	-0.791	0.118	0.200
0.458	-0.103	-0.089	0.074	0.022	0.458	-0.655	0.044	1.013	0.953
0.200	2.365	0.981	-0.089	0.883	-1.151	0.007	0.981	0.303	1.308
-0.673	-0.443	-0.687	0.643	-0.237	-0.067	-0.621	0.325	0.015	-1.094
1.345	0.148	0.177	-0.153	-0.698	1.227	0.995	-0.503	-0.792	

RECEIVER OUTPUT IN DB

SAMPLE SPACING = 29.2969

0.	0.	-0.00	-0.01	-0.02	-0.04
146.48	-0.08	-0.13	-0.16	-0.17	-0.15
292.97	-0.11	-0.04	0.01	0.03	0.00
439.45	-0.08	-0.20	-0.34	-0.46	-0.52
585.94	-0.50	-0.43	-0.34	-0.27	-0.25
732.42	-0.31	-0.43	-0.58	-0.71	-0.76
878.91	-0.71	-0.53	-0.28	-0.02	0.21
1025.39	0.35	0.40	0.38	0.30	0.23
1171.87	0.20	0.23	0.30	0.38	0.43
1318.36	0.40	0.29	0.10	-0.15	-0.40
1464.84	-0.60	-0.69	-0.65	-0.52	-0.35
1611.33	-0.20	-0.10	-0.08	-0.05	-0.06
1757.81	-0.05	0.00	0.09	0.21	0.31
1904.30	0.39	0.40	0.34	0.23	0.08
2050.78	-0.07	-0.17	-0.22	-0.20	-0.13
2197.27	-0.03	0.06	0.11	0.12	0.08
2343.75	-0.00	-0.12	-0.24	-0.37	-0.47
2490.23	-0.55	-0.58	-0.56	-0.46	-0.35
2636.72	-0.19	-0.03	0.10	0.18	0.19
2783.20	0.15	0.08	-0.04	-0.14	-0.22
2929.69	-0.30	-0.44	-0.71	-1.20	-1.97
3076.17	-3.10	-4.46	-6.68	-9.22	-12.30
3222.66	-15.94	-20.14	-24.89	-30.22	-36.30
3369.14	-43.71	-53.90	-58.76	-59.08	-71.63
3515.62	-63.11	-59.43	-61.83	-75.45	-66.82
3662.11	-62.77	-64.19	-67.70	-65.92	-70.98
3808.59	-71.74	-69.94	-61.11	-61.62	-65.19
3955.08	-70.84	-60.98	-66.25	-65.02	-65.95
4101.56	-69.18	-64.99	-61.62	-59.51	-59.89
4248.05	-60.23	-61.94	-65.49	-61.13	-67.70
4394.53	-56.23	-57.17	-61.20	-69.46	-63.36
4541.02	-64.43	-69.46	-70.81	-79.78	-69.07
4687.50	-65.56	-69.63	-63.79	-67.73	-85.03
4833.98	-68.14	-64.58	-65.56	-73.42	-75.97
4980.47	-70.58	-70.54	-70.80	-69.80	-65.42
5126.95	-65.04	-64.45	-64.53	[illegible]	[illegible]
5273.44	-63.02	-65.86	-62.71	-69.53	-63.14
5419.92	-64.71	-65.37	-64.72	-65.01	-73.47
5566.41	-72.39	-64.77	-63.51	[illegible]	-81.59
5712.89	-70.06	-65.64	-68.03	[illegible]	-67.33
5859.37	-64.75	-64.96	-63.46	[illegible]	-60.91
6005.86	-61.75	[illegible]	-68.84	[illegible]	[illegible]
6152.34	-73.84	-68.81	-64.77	[illegible]	-63.91
6298.83	[illegible]	-65.09	-63.96	[illegible]	[illegible]
6445.31	-77.64	-63.41	-60.96	[illegible]	-62.37
6591.80	-65.98	-60.83	-72.14	-60.08	-69.93

6738.28	-67.07	-70.40	-67.57	-61.35	-60.46
6884.77	-62.97	-66.11	-61.71	-59.22	-60.04
7031.25	-65.14	-83.60	-65.69	-62.72	-63.62
7177.73	-67.68	-73.96	-73.26	-67.27	-64.19
7324.22	-63.10	-63.03	-66.90	-72.43	-77.48
7470.70	-75.99	-73.21	-75.99	-77.48	-72.43
7617.19	-66.90	-63.03	-63.10	-64.19	-67.27
7763.67	-73.26	-75.96	-67.68	-63.52	-62.72
7910.16	-65.69	-83.60	-65.14	-60.04	-59.22
8056.64	-61.71	-66.11	-62.97	-60.46	-61.55
8203.12	-67.57	-70.40	-67.07	-65.93	-69.08
8349.61	-72.14	-69.33	-65.90	-62.37	-60.27
8496.09	-60.56	-64.41	-77.64	-66.73	-62.91
8642.58	-63.86	-66.05	-66.79	-63.21	-63.28
8789.06	-64.77	-68.01	-73.24	-84.34	-76.12
8935.55	-68.64	-64.28	-61.75	-60.91	-61.70
9082.03	-63.46	-64.36	-64.75	-67.33	-72.46
9228.52	-68.83	-66.64	-70.06	-81.99	-66.76
9375.00	-63.51	-64.77	-72.39	-73.47	-66.01
9521.48	-64.72	-65.37	-64.71	-63.14	-62.53
9667.97	-62.71	-62.86	-63.02	-63.49	-64.77
9814.45	-65.53	-68.45	-65.04	-65.42	-67.50
9960.94	-70.80	-70.84	-70.58	-73.97	-70.62
10107.42	-65.56	-64.58	-68.14	-85.03	-67.73
10253.91	-63.79	-63.63	-65.56	-69.07	-79.72
10400.39	-70.81	-63.46	-61.43	-63.36	-67.46
10546.87	-61.20	-57.17	-56.23	-57.70	-61.13
10693.36	-63.49	-61.94	-60.23	-59.29	-59.51
10839.84	-61.62	-64.39	-60.18	-65.05	-65.02
10986.33	-66.25	-60.98	-70.84	-68.19	-61.62
11132.81	-61.11	-63.94	-71.74	-70.98	-68.22
11279.30	-67.70	-64.19	-62.77	-66.32	-75.65
11425.78	-61.83	-50.43	-63.11	-71.43	-59.08
11572.27	-56.76	-55.00	-43.71	-36.30	-30.22
11718.75	-24.89	-20.14	-15.94	-12.30	-9.22
11865.23	-6.68	-4.66	-3.10	-1.97	-1.20
12011.72	-0.71	-0.44	-0.30	-0.22	-0.14
12158.20	-0.04	0.06	0.15	0.19	-0.18
12304.69	0.10	-0.03	-0.19	-0.35	-0.48
12451.17	-0.56	-0.58	-0.55	-0.47	-0.37
12597.66	-0.24	-0.12	-0.00	0.08	0.12
12744.14	0.11	0.06	-0.03	-0.13	-0.20
12890.62	-0.22	-0.17	-0.07	0.08	0.23
13037.11	0.34	0.40	0.39	0.31	0.21
13183.59	0.09	0.00	-0.05	-0.06	-0.05
13330.08	-0.06	-0.10	-0.20	-0.35	-0.32
13476.56	-0.65	-0.69	-0.60	-0.40	-0.15
13623.05	0.10	0.29	0.40	0.43	0.38
13769.53	0.30	0.23	0.20	0.23	0.30
13916.02	0.38	0.40	0.35	0.21	-0.02
14062.50	-0.28	-0.53	-0.71	-0.76	-0.71
14208.98	-0.58	-0.43	-0.31	-0.29	-0.27
14355.47	-0.34	-0.43	-0.50	-0.52	-0.46
14501.95	-0.34	-0.20	-0.08	0.00	0.03
14648.44	0.01	-0.04	-0.11	-0.15	-0.17
14794.92	-0.16	-0.13	-0.08	-0.04	-0.02
14941.41	-0.01	-0.00	0.	0.	0.

INTEGRAL SQUARED NOISE = 0.248

SIGNAL TO NOISE RATIO = 11.88 DB

SIGNAL TO NOISE POWER OUT = 29.189 DB

4. FM AND PM SIMULATION

```
      DIMENSION WSSB(100),WREP(100),ANLOG(100),X(100),XR(300),DIF(100),
     * XI(300),XR2(400),XI2(400),XR3(900),XI3(900),XR4(150),XI4(150),
     * XR5(2000),XI5(2000),XR6(400),XI6(400),XR7(1000),XI7(1000),
     * XR8(350),XI8(350),XR9(400),XI9(400),XOUT(200),FTM(512),XF(512)
      DIMENSION XT(50),E1(220),F1(512),F2(512)
      COMPLEX FT(2,512),C
      DATA LF,L/10,8/
      DATA MSIG,MS,MR,MA/5,45,40,20/
      DATA NS,NSB,NRF,NA/30,91,81,41/
C NS=NO. OF TERMS IN INPUT SIGNAL = 2*MSIG+1
      DATA A1,A3,A4,A8,A9,A0/1.284,2.525,1.0,5.64,7.07,9.0//
C NSB=NO. OF TERMS IN DSB FILTER = 2*MS+1
C NRF=NO. OF TERMS IN REP FILTER = 2*MR+1
C NA=NO. OF TERMS IN ANALOG FILTER = 2*MA+1
      DATA FZS,FZR,FZA/.4266,.1258,.0312/
      DATA SGN/10,65/
      DATA CF/0.5/
      DATA ALPHA,BETA,H/0.0,0.517,5.92/
C ALPHA - PREDISTORTION COEFICIENT
C BETA - MODULATION INDEX
      PI=3.14159265
CCALCULATE TAP WEIGHTS
      DO 50 I=1,MS
      FI=I
      WIN=.54+.46*COS(PI*FI/48.)
      IMS=MS+1+I
      WSSB(IMS)=WIN*SIN(FZS*PI*FI)/(PI*FI*FZS)
      CALL RND(WSSB(IMS),1,0,LF)
      JMS=MS+1-I
50    WSSB(JMS)=WSSB(IMS)
      WSSB(MS+1)=1.0
      CALL RND(WSSB(MS+1),1,0,LF)
      FMR=MR
      DO 60 I=1,MR
      FI=I
      WIN=.54+.46*COS(PI*FI/FMR)
      IMR=MR+1+I
      WREP(IMR)=WIN*SIN(FZR*PI*FI)/(PI*FI*FZR)
      CALL RND(WREP(IMR),1,0,LF)
      JMR=MR+1-I
60    WREP(JMR)=WREP(IMR)
      WREP(MR+1)=1.0
      CALL RND(WREP(MR+1),1,0,LF)
      FMA=MA
      DO 70 I=1,MA
      FI=I
      WIN=.54+.46*COS(PI*FI/FMA)
      IMA=MA+1+I
      ANLOG(IMA)=WIN*SIN(FZA*PI*FI)/(PI*FI*FZA)
      JMA=MA+1-I
      ANLOG(JMA)=ANLOG(IMA)
70    CONTINUE
      ANLOG(MA+1)=1.0
      A=1
      DO 75 I=1,MS
      FI=I
```

```
      IMS=MS+1+I
      WIN=.54+.46*COS(PI*FI/51.)
      A=-A
      DIF(IMS)=WIN*A/FI
      CALL RND(DIF(IMS),1.0,LF)
      JMS=MS+1-I
   75 DIF(JMS)=-DIF(IMS)
      DIF(MS+1)=0.0
C LOAD 50 PERCENT RC PULSE
      A=6.0*PI/15.0
      DO 80 I=1,MSIG
      IMSIG=MSIG+1+I
      FI=I
      X(IMSIG)=SIN(A*FI)*COS(.5*A*FI)/(A*FI*(1.-(6.*FI/15.)**2))
      CALL RND(X(IMSIG),1.0,7)
      JMSIG=MSIG+1-I
   80 X(JMSIG)=X(IMSIG)
      X(MSIG+1)=1.0
      CALL RND(X(MSIG+1),1.0,7)
      XMAX=1.0
      DO 90 J=1,50
      DO 90 I=1,11
      IJ=I+J-1
      XT(IJ)=X(I)+XT(IJ)
   90 IF(XT(IJ).GT.XMAX) XMAX=XT(IJ)
      DO 95 I=1,50
   95 X(I)=CF*XT(I)/(XMAX*.895)
      PRINT 96,(X(I),I=1,42)
   96 FORMAT(1H1,16HINPUT WAVE FORM  /(10F10.3))
      DO 110 I=1,NS
  110 XR(I)=X(I)
C CONVOLVE WITH FIRST FILTER
      DO 112 I=1,NS
C SHIFT ALL TERMS BY NSB
      NSI=NS+1-I
      NSBI=NSB+NS-I
      XR(NSBI)=XR(NSI)
  112 XI(NSBI)=XI(NSI)
      NSB1=NS-1
      DO 113 I=1,NSB1
      XR(I)=0.0
  113 XI(I)=0.0
C NF=NO. OF TERMS OUT OF FIRST FILTER
      NF=NSB+NS-1
      DO 115 I=1,NF
      DO 114 J=1,NSB
      IJ=I+J-1
      XR2(I)=XR2(I)+WSSB(J)*XR(IJ)
  114 XI2(I)=0.0
      CALL RND(XR2(I),A1,L)
  115 CALL RND(XI2(I),A1,L)
      PRINT 116
  116 FORMAT(1H0,30HWAVE FORM OUT OF FIRST FILTER    )
      PRINT 13,(XR2(I),I=1,NF)
C EFFECTIVELY FILL WITH ZEROS AND CONVOLVE WITH 7 KC LOW-PASS
C REPETITION FILTER, OUTPUT OF FILTER IS AT 120 KC
      DO 150 I=1,NF
```

```
C SHIFT ALL TERMS BY NRF
      NFI=NF+1-I
      NRFI=NRF+NF-I
      XI2(NRFI)=XI2(NFI)
150   XR2(NRFI)=XR2(NFI)
      NRF1=NRF-1
      DO 160 I=1,NRF1
      XR2(I)=0.0
160   XI2(I)=0.0
C CALCULATE 120 KC OUTPUT
      J=0
      DO 170 L1=1,NF
      DO 170 K=1,8
      J=J+1
      DO 170 I=1,11
      LT=L1-I+92
      KI=8*(I-1)+K
      XI3(J)=XI3(J)+WREP(KI)*XI2(LT)
170   XR3(J)=XR3(J)+WREP(KI)*XR2(LT)
C J IS NUMBER OF OUTPUT SAMPLES FROM REP FILTER
      K2=J
      DO 180 I=1,K2
180   CALL RND(XR3(I),A3,L)
C FM MODULATOR
      DO 185 I=1,K2
      XI3(I)=0.0
      XRF=XR3(I)*BETA/H   +XRF
      XF(I)=XRF
185   CONTINUE
      DO 186 I=1,K2
      FI=I
      XR3(I)=CS(XF(I),LF)*(.54+.46*COS(PI*(FI-480.)/480.))
186   XI3(I)=SN(XF(I),LF)*(.54+.46*COS(PI*(FI-480.)/480.))
C PUT SIGNAL THROUGH INTERPOLATION FILTER AND SIMULATED
C ANALOG FILTER
      K=0
      DO 190 I=1,K2
      DELI=(XI3(I+1)-XI3(I))/32.0
      DELR=(XR3(I+1)-XR3(I))/32.0
      DO 190 J=1,32
      K=K+1
      K1=MOD(K,100)+1
C K IS XR4 INDEX
C K1 IS XR4 CYLIC INDEX
      FJ=J-1
      XI4(K1)=XI3(I)+(FJ-1.)*DELI
      XR4(K1)=XR3(I)+(FJ-1.)*DELR
      CALL RND(XR4(K1),A4,L)
      CALL RND(XI4(K1),A4,L)
      IF(MOD(K,16).EQ.0) GO TO 200
190   CONTINUE
      GO TO 220
200   CONTINUE
C CALCULATE OUTPUT OF ANALOG FILTER
C M IS XR5 INDEX
      M=M+1
      DO 210 L1=1,NA
```

```
      N=MOD(74+(M-1)*16+L1,100)+1
      XI5(M)=XI5(M)+ANLOG(L1)*XI4(N)
210   XR5(M)=XR5(M)+ANLOG(L1)*XR4(N)
      GO TO 190
220   CONTINUE
      M1=M/2+1
      M2=M/5+1
      DO 230 I=1,M1
      XI6(I)=XI5(2*I-1)
      XR6(I)=XR5(2*I-1)
230   CONTINUE
      DO 240 I=1,M2
      XI7(I)=XI5(5*I-4)
240   XR7(I)=XR5(5*I-4)
      GO TO 291
241   CONTINUE
C XR7 IS INPUT TO FFT TO LOOK AT OUTPUT SPECTRUM OF TRANSMITTER
C XR6 IS INPUT TO RECEIVER
C ADD NOISE AND SHIFT BY NRF-6
      FM1=2*M1
      DO 249 I=1,M1
249   XS=XS+XR6(I)*XR6(I)+XI6(I)*XI6(I)
      SGN=SQRT(XS/FM1)
      DO 250 I=1,M1
      CALL GAUSS(Z,SGM,SGN)
      ZT=ZT+Z*Z
      JM=M1+NRF-8-I
      IM=M1+1-I
      XR6(JM)=XR6(IM)+Z
      CALL GAUSS(Z,SGM,SGN)
      ZT=ZT+Z*Z
      XI6(JM)=XI6(IM)+Z
250   CONTINUE
      IF(ZT.EQ.0.) GO TO 251
      SG=10.0*ALOG10(XS/ZT)
251   CONTINUE
C SIGNAL GOES INTO REPETITION FILTER IN STEPS OF 8 INPUT SAMPLES
C OUTPUT IS AT 15 KC
      N8=NRF-9
      DO 260 I=1,N8
      XR6(I)=0.0
260   XI6(I)=0.0
C CALCULATE OUTPUT OF RECEIVER REPETITION FILTER
      M3=M1/8+1
      DO 280 K=1,M3
      DO 270 L1=1,NRF
      LK=L1+8*(K-1)
      NL=NRF+1-L1
      XI8(K)=XI8(K)+WREP(NL)*XI6(LK)
270   XR8(K)=XR8(K)+WREP(NL)*XR6(LK)
      CALL RND(XR8(K),A8,L)
280   CALL RND(XI8(K),A8,L)
      PRINT 13,(XR8(I),I=1,M3)
      PRINT 13,(XI8(I),I=1,M3)
      DO 281 I=1,M3
      NI=NSB+M3-I
      MI=M3+1-I
```

```
      XR8(NI)=XR8(MI)
 281  XI8(NI)=XI8(MI)
      NX=NS8-1
      DO 282 I=1,NX
      XR8(I)=0.0
 282  XI8(I)=0.0
      M4=M3+NS8-1
      DO 283 I=1,M4
      DO 284 J=1,NS8
      NJ=NS8+1-J
      IJ=I+J-1
      XR9(I)=XR9(I)+DIF(NJ)*XR8(IJ)
 284  XI9(I)=XI9(I)+DIF(NJ)*XI8(IJ)
      CALL RND(XR9(I),A9,L)
 283  CALL RND(XI9(I),A9,L)
C FM DEMODULATOR
      DO 290 I=1,M4
      IMS=MS+I
      A=XI8(IMS)*XR9(I)
      CALL RND(A,A9,LF)
      B=XR8(IMS)*XI9(I)
      CALL RND(B,A9,LF)
      F=XR8(IMS)**2
      CALL RND(F,A9,LF)
      D=XI8(IMS)**2
      CALL RND(D,A9,LF)
      E=F+D
      E1(I)=F
      IF(E.LT.0.5) E=1.0
 290  XR9(I)=(B-A)/E
      PRINT 13,(E1(I),I=1,M4)
      FM4=M4
      DO 293 I=1,M4
 293  ET=ET+E1(I)
      EA=ET/FM4
      DO 294 I=1,M4
 294  SM=SM+(E1(I)-EA)**2
      SM=SM/FM4
      PRINT 12
 12   FORMAT(1H0,26HWAVE FORM OUT OF FM DEMOD     )
      PRINT 13,(XR9(I),I=1,M4)
 13   FORMAT(1H ,10F12.3)
      DO 296 I=1,M4
      MI=M4+1-I
      NI=NS8+M4-I
 296  XR9(NI)=XR9(MI)
      DO 297 I=1,NX
 297  XR9(I)=0.0
      M5=M4+NS8-1
      PRINT 13,(XR9(I),I=1,M5)
      PRINT 13,WSSB
      DO 299 I=1,M5
      XOUT(I)=0.0
      DO 298 J=1,NS8
      IJ=I+J-1
 298  XOUT(I)=XOUT(I)+WSSB(J)*XR9(IJ)
 299  CONTINUE
```

```
      PRINT 13, (XOUT(I),I=1,M5)
      GO TO 341
C
C THIS SECTION CALCULATES AND PRINTS OUT SPECTRUM OF
C TRANSMITTER AND OUTPUT
C TRANSMITTER OUTPUT
291   CONTINUE
      DO 300 I=1,M2
300   FT(1,I)=CMPLX(XR7(I),XI7(I))
      CALL FFT(FT,9,-1,)
      DO 310 I=1,512
      G=CABS(FT(2,I))
      IF(G.LE.0.00001) G=0.00001
310   FTM(I)=20.0*ALOG10(G)
      B=FTM(1)
      DO 315 I=1,512
315   FTM(I)=FTM(I)-B
      PRINT 320
320   FORMAT(1H1,34HSPECTRUM IN DB AFTER ANALOG FILTER )
      SMPC= 48000.0/512.0
      PRINT 330, SMPC
330   FORMAT(1H ,16HSAMPLE SPACING - ,F10.4)
      DO 340 I=1,103
      J=I-1
      FJ=J
      CI=FJ*5.0*SMPC
      K1=5*J+1
      K2=K1+4
      PRINT 350, CI,(FTM(K3),K3=K1,K2)
350   FORMAT(1H ,(6F12.2))
340   CONTINUE
      GO TO 241
C
341   CONTINUE
C RECEIVER OUTPUT
      DO 360 I=1,512
360   FT(1,I)=CMPLX(0.,0.)
      DO 370 I=1,M5
370   FT(1,I)=CMPLX(XOUT(I),0.)
      CALL FFT(FT,9,-1,)
      DO 380 I=1,512
      F1(I)=CABS(FT(2,I))
      IF(F1(I).LT.0.00001) F1(I)=0.00001
380   FTM(I)=20.0*ALOG10(F1(I))
      B=FTM(1)
      B1=F1(1)
      DO 381 I=1,512
      F1(I)=F1(I)/B1
381   FTM(I)=FTM(I)-B
      SMPC=15000./512.
      PRINT 385
385   FORMAT(1H1,30HRECEIVER OUTPUT SPECTRUM IN DB   )
      PRINT 390,SMPC
390   FORMAT(1H ,16HSAMPLE SPACING = ,F10.4)
      DO 400 I=1,52
      J=I-1
      FJ=J
```

```
      CI=FJ*5,0*SMPC
      K1=5*J+1
      K2=K1+4
400   PRINT 350,CI,(FTM(K3),K3=K1,K2)
      PRINT 900,(XOUT(I),I=1,M5)
900   FORMAT(1H0,23HTIME RESPONSE OF OUTPUT  /(10F12.3))
      PRINT 425 ,SG
425   FORMAT(1H0,44HSIGNAL TO NOISE RATIO AT INPUT TO RECEIVER = ,F10.2
     * ,3H DB )
C
      READ 435,(F2(I),I=1,M5)
      DO 426 I=178,159
      ER2=ER2+F2(I)*F2(I)
426   ER=ER+(XOUT(I)-F2(I))**2
      SGA=10.0*ALOG10(ER2/ER)
      PRINT 427,SGA
427   FORMAT(1H0,27HSIGNAL TO NOISE POWER OUT =   ,F10.3,3H DB )
435   FORMAT(10F8.5)
999   CONTINUE
      STOP
      END
```

INPUT WAVE FORM

0.004	0.008	-0.017	-0.029	0.118	0.388	0.529	0.517	0.491	0.496
0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500
0.496	0.491	0.517	0.529	0.388	0.118	-0.029	-0.017	0.008	0.004
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.								

WAVE FORM OUT OF FIRST FILTER

0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	-0.010	0.010	0.	0.	0.	0.010	0.	0.	0.
0.010	0.010	0.	-0.010	0.	0.010	0.010	-0.010	-0.010	0.
0.020	0.	-0.030	-0.020	0.040	0.091	-0.080	-0.101	0.	-0.
0.819	1.173	1.274	1.203	1.122	1.122	1.183	1.213	1.188	[illegible]
1.142	1.183	1.213	1.183	1.122	1.122	1.203	1.274	1.173	0.
0.354	0.	-0.101	-0.030	0.091	0.040	-0.020	-0.030	0.	0.
0.010	-0.010	-0.010	0.010	0.010	0.	-0.010	0.	0.010	0.
0.	0.	0.	0.010	0.	0.	0.	0.010	0.010	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.

SPECTRUM IN DB AFTER ANALOG FILTER
SAMPLE SPACING = 93.7500

0.	0.	-1.78	5.43	-1.41	-9.55
468.75	-25.68	-10.55	-6.45	-6.00	-5.22
937.50	-6.02	-10.10	-14.90	-23.24	-10.20
1406.25	-5.56	-2.63	-0.73	0.46	0.99
1875.00	1.20	0.87	0.12	-0.89	-2.79
2343.75	-4.95	-7.90	-13.04	-20.22	-32.30
2812.50	-20.52	-17.07	-17.38	-17.97	-19.82
3281.25	-26.59	-32.90	-41.11	-29.51	-27.38
3750.00	-25.72	-25.81	-26.55	-28.78	-32.08
4218.75	-42.55	-40.77	-40.27	-30.46	-30.24
4687.50	-31.64	-28.94	-30.64	-33.65	-34.72
5156.25	-36.70	-54.95	-47.74	-54.05	-42.24
5625.00	-48.93	-45.82	-42.64	-49.03	-44.82
6093.75	-63.81	-51.46	-47.42	-48.75	-55.98
6562.50	-44.31	-50.12	-46.71	-51.75	-47.18
7031.25	-47.43	-52.27	-48.58	-47.00	-52.91
7500.00	-55.72	-60.52	-50.00	-52.79	-62.33
7968.75	-59.13	-55.24	-48.48	-53.84	-57.46
8437.50	-50.27	-46.92	-49.23	-52.82	-46.94
8906.25	-47.53	-53.18	-52.97	-46.45	-47.97
9375.00	-62.69	-48.29	-49.72	-51.41	-59.18
9843.75	-51.92	-54.89	-56.81	-65.40	-61.36
10312.50	-51.87	-53.97	-66.26	-62.27	-52.71
10781.25	-51.87	-55.36	-52.16	-51.55	-51.82
11250.00	-62.24	-57.57	-50.45	-58.72	-61.24
11718.75	-58.78	-50.18	-56.05	-58.81	-59.27
12187.50	-57.29	-58.30	-63.18	-62.88	-53.28
12656.25	-54.74	-58.68	-54.66	-53.47	-53.43
13125.00	-65.08	-57.27	-54.23	-58.16	-63.63
13593.75	-59.10	-54.64	-58.07	-57.20	-60.83
14062.50	-60.76	-52.95	-53.66	-60.33	-58.84
14531.25	-51.78	-72.00	-57.07	-51.21	-51.93
15000.00	-65.93	-54.43	-53.43	-54.47	-65.04
15468.75	-56.29	-51.27	-57.64	-60.17	-55.73
15937.50	-53.97	-61.89	-57.19	-59.03	-55.03
16406.25	-55.92	-64.51	-52.50	-57.85	-54.62
16875.00	-61.96	-57.51	-55.25	-57.59	-65.91
17343.75	-52.45	-55.63	-60.12	-68.22	-62.01
17812.50	-56.07	-78.84	-55.97	-58.94	-65.96
18281.25	-62.76	-55.83	-57.02	-55.15	-55.54
18750.00	-74.54	-59.89	-53.01	-61.45	-62.13
19218.75	-56.97	-55.64	-57.39	-55.99	-53.56
19687.50	-62.76	-74.60	-60.91	-61.10	-61.42
20156.25	-61.13	-57.14	-57.93	-55.67	-72.33
20625.00	-60.39	-61.45	-52.25	-62.83	-57.85
21093.75	-64.16	-55.33	-58.91	-59.09	-56.36
21562.50	-58.16	-64.97	-62.47	-58.74	-64.85
22031.25	-61.62	-60.76	-61.35	-61.02	-59.10
22500.00	-61.46	-64.02	-55.31	-57.11	-56.64
22968.75	-55.03	-63.12	-64.50	-67.54	-58.44
23437.50	-57.27	-71.47	-66.52	-59.06	-67.08
23906.25	-54.25	-60.37	-51.82	-55.25	-56.96
24375.00	-49.02	-56.66	-58.81	-66.47	-60.87
24843.75	-61.54	-55.39	-53.72	-66.48	-57.35
25312.50	-56.55	-60.71	-55.45	-61.98	-54.80
25781.25	-71.36	-54.96	-58.10	-58.29	-59.49
26250.00	-56.53	-63.19	-63.16	-61.89	-57.34
26718.75	-64.63	-60.28	-60.87	-54.09	-56.01
27187.50	-55.54	-59.59	-60.31	-53.88	-57.75

27656,25	-62,58	-54,81	-50,86	-60,50	-61,83				
28125,00	-54,46	-54,81	-73,39	-58,83	-57,80				
28593,75	-60,89	-59,53	-58,31	-57,13	-55,33				
29062,50	-68,22	-69,41	-60,42	-82,77	-56,62				
29531,25	-58,53	-58,25	-55,50	-65,12	-79,11				
30000,00	-59,15	-54,97	-60,17	-50,79	-53,72				
30468,75	-53,78	-62,16	-56,62	-58,27	-57,95				
30937,50	-64,52	-68,93	-55,78	-58,43	-78,64				
31406,25	-62,30	-53,58	-54,67	-61,90	-67,45				
31875,00	-52,29	-54,69	-68,71	-60,81	-58,01				
32343,75	-50,55	-65,77	-55,39	-53,95	-57,30				
32812,50	-56,90	-62,20	-57,66	-57,33	-61,88				
33281,25	-59,46	-81,80	-54,09	-59,33	-68,55				
33750,00	-52,67	-53,81	-59,84	-58,82	-51,03				
34218,75	-63,19	-63,23	-52,58	-48,72	-68,06				
34687,50	-54,59	-53,04	-51,72	-68,79	-58,53				
35156,25	-57,80	-61,63	-59,21	-72,54	-68,57				
35625,00	-57,73	-51,85	-62,01	-58,63	-52,83				
36093,75	-48,36	-66,93	-49,03	-49,49	-52,44				
36562,50	-54,97	-51,71	-50,54	-61,73	-56,00				
37031,25	-53,34	-52,96	-59,92	-59,70	-55,16				
37500,00	-67,29	-54,40	-58,57	-55,79	-51,48				
37968,75	-53,31	-50,63	-50,65	-47,66	-54,92				
38437,50	-56,92	-50,24	-47,54	-57,51	-64,71				
38906,25	-48,45	-50,35	-62,76	-55,36	-50,00				
39375,00	-55,17	-60,13	-64,99	-50,59	-51,42				
39843,75	-62,13	-65,70	-55,76	-47,29	-52,04				
40312,50	-52,80	-44,00	-46,49	-55,45	-52,15				
40781,25	-48,90	-47,56	-53,44	-51,64	-48,87				
41250,00	-48,33	-81,04	-52,67	-53,97	-49,99				
41718,75	-59,06	-75,50	-47,55	-49,41	-48,82				
42187,50	-54,15	-44,34	-46,52	-47,37	-45,73				
42656,25	-44,27	-44,48	-53,14	-49,26	-44,97				
43125,00	-44,96	-53,29	-44,08	-48,91	-47,77				
43593,75	-66,93	-55,27	-46,01	-47,26	-47,52				
44062,50	-47,19	-44,77	-37,97	-68,90	-44,93				
44531,25	-48,90	-37,34	-33,90	-39,49	-36,27				
45000,00	-45,96	-33,10	-29,20	-33,20	-30,20				
45468,75	-37,67	38,84	-35,99	-28,11	-28,45				
45937,50	-27,89	-27,32	-34,08	-40,40	-30,69				
46406,25	-27,12	-20,77	-22,12	-23,85	-21,38				
46875,00	-38,97	-24,55	-20,42	-15,18	-11,82				
47343,75	-10,57	-10,92	-10,91	-18,86	-11,68				
47812,50	1,99	8,78	0,	0,	0,				
0.044	-0,044	0,222	-0,489	3,553	9,948	12,746	12,612	13,190	14,34
20.872	16,032	19,940	22,604	18,608	22,693	25,091	33,218	32,419	35,21
29.754	41,523	42,855	44,898	49,739	51,959	53,957	58,976	64,083	65,63
73.098	75,763	80,914	82,513	82,024	90,640	92,985	92,327	100,410	104,62
103.474	110,491	115,109	112,622	119,906	126,523	127,544	130,386	114,080	48,05.
-69.190	-138,113	-113,733	-21,938	85,666	148,727	110,002	1,554	-105,384	-147,080
-101.076	16,032	124,391	157,787	93,260	-86,238	-135,804	-152,102	-125,279	-118,08
-123.503	-121,415	-120.483	-118,573	-116,131	-114,621	-114,354	-112,534	-111,867	-107,95
-105.206	-102,630	-99,344	-99,166	-92,327	-92,949	-89,307	-87.309	-83,579	-80,64
-75.008	-73,586	-69,678	-68,302	-68,435	-63,905	-58,487	-53,325	-55,467	-48,54
-44.409	-45,875	-41,834	-37,704	-38,103	-38,681	-30,953	-29,868	-25,716	-25,40.
-27.179	-20,828	-17,542	-20,562	-18,563	-17,142	-15,677	-12,124	-11,102	-11,85.
-10.703									
-0.044	0,089	-0.311	0.666	-5,107	-14,344	-15,854	-11.191	-15,987	-6,57
-11.014	-8,171	2.443	-0,933	3,686	-2,709	-0,533	0,888	0,400	1,06
-3.952	0,178	-1,998	-3,420	-0,222	2,753	2,842	0,973	0,577	1,46

RECEIVER OUTPUT SPECTRUM IN DB
SAMPLE SPACING = 29.2969

0.	0.	-0.19	-0.74	-1.51	-2.27
146.48	-2.70	-2.66	-2.32	-2.01	-1.97
292.97	-2.31	-3.05	-4.18	-5.60	-7.06
439.45	-8.09	-8.40	-8.40	-8.66	-9.45
585.94	-10.83	-12.83	-15.73	-20.56	-32.50
732.42	-28.02	-22.16	-21.22	-22.76	-24.00
878.91	-21.44	-18.53	-16.84	-15.11	-14.04
1025.39	-13.41	-13.30	-13.59	-14.10	-14.71
1171.87	-15.46	-18.33	-17.15	-17.89	-18.86
1318.36	-20.15	-20.91	-20.55	-20.37	-21.61
1464.84	-24.88	-27.87	-24.95	-22.19	-20.65
1611.33	-20.13	-19.72	-19.99	-21.60	-25.17
1757.81	-26.82	-21.83	-17.60	-15.14	-13.80
1904.30	-13.41	-13.99	-15.70	-19.01	-25.44
2050.78	-32.15	-23.69	-20.27	-19.30	-19.82
2197.27	-20.82	-20.91	-20.45	-20.91	-22.48
2343.75	-22.50	-20.03	-18.45	-18.82	-21.45
2490.23	-25.73	-25.06	-22.74	-22.73	-23.52
2636.72	-22.01	-19.89	-19.21	-20.02	-21.54
2783.20	-22.60	-24.03	-28.73	-35.67	-25.16
2929.69	-2[illegible].73	-21.92	-25.21	-26.92	-23.11
3076.17	-21.69	-23.20	-27.24	-30.08	-29.78
3222.66	-32.19	-40.71	-42.96	-38.21	-39.82
3369.14	-45.87	-58.40	-78.50	-78.50	-67.56
3515.62	-69.68	-78.50	-78.50	-78.50	-73.75
3662.11	-74.68	-78.50	-78.50	-78.50	-78.50
3808.59	-78.50	-78.50	-78.50	-78.50	-78.50
3955.08	-74.01	-72.29	-74.03	-77.27	-75.21
4101.56	-73.51	-74.03	-74.15	-71.81	-70.09
4248.05	-71.04	-77.14	-78.50	-77.39	-78.50
4394.53	-78.50	-78.50	-78.50	-73.38	-71.53
4541.02	-76.07	-78.50	-76.54	-76.15	-78.50
4687.50	-78.50	-78.50	-78.50	-78.50	-78.50
4833.98	-78.50	-74.87	-73.46	-78.50	-78.50
4980.47	-70.18	-69.26	-73.82	-78.50	-78.50
5126.95	-78.50	-75.77	-78.03	-78.50	-78.50
5273.44	-76.01	-70.65	-72.63	-78.50	-69.69
5419.92	-68.59	-74.01	-78.50	-78.50	-74.69
5566.41	-70.14	-71.57	-78.50	-78.50	-78.50
5712.89	-78.50	-78.50	-78.50	-78.50	-78.50
5859.37	-73.64	-68.92	-68.93	-67.54	-62.80
6005.86	-60.71	-61.53	-64.29	-66.41	-69.04
6152.34	-78.50	-72.54	-66.24	-63.34	-61.58
6298.83	-61.25	-62.72	-66.01	-70.80	-77.30
6445.31	-78.50	-78.41	-74.71	-75.23	-77.64
6591.80	-78.50	-78.29	-76.82	-75.57	-75.69
6738.28	-76.30	-74.81	-74.91	-78.41	-78.60
6884.77	-78.50	-73.07	-70.30	-72.22	-78.50
7031.25	-74.66	-72.00	-75.06	-78.80	-78.50
7177.73	-78.50	-78.50	-78.50	-78.50	-78.50
7324.22	-78.50	-74.18	-73.19	-74.00	-78.50
7470.70	-78.37	-74.35	-78.37	-78.50	-74.00

TIME RESPONSE OF OUTPUT

0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.000	0.000	0.001	0.00

0.001	-0.000	-0.000	0.000	0.001	0.002	-0.000	-0.000	0.001	0.00
-0.000	-0.002	-0.002	0.003	0.009	-0.008	-0.003	-0.001	0.004	[illegible]
-0.006	-0.007	0.009	0.011	0.003	-0.018	-0.007	0.009	0.019	-0.00
-0.022	-0.004	0.021	0.020	-0.021	-0.033	-0.011	0.048	0.011	-0.00
0.025	0.333	0.704	0.849	0.620	0.198	-0.071	0.091	0.340	0.44
0.685	0.463	0.108	-0.107	-0.207	-0.179	0.027	0.119	0.109	-0.04
-0.137	-0.093	0.037	0.133	0.127	0.053	-0.007	-0.026	-0.029	-0.03
-0.044	-0.003	0.081	0.008	0.080	-0.058	-0.076	-0.034	0.018	0.01
0.001	0.029	0.078	0.089	-0.071	-0.150	0.041	0.607	1.349	1.01
2.092	1.976	1.821	1.819	1.933	2.010	1.961	1.861	1.849	1.04
2.001	1.942	1.821	1.816	1.984	2.118	1.909	1.279	0.903	-0.02
-0.137	-0.012	0.084	0.091	-0.024	-0.037	-0.003	0.008	-0.010	-0.01
0.023	0.049	0.026	-0.019	-0.029	-0.006	0.005	-0.013	-0.013	0.03
0.000	0.045	-0.048	-0.085	-0.016	0.100	0.122	0.350	-0.013	-0.00
0.002	-0.045	-0.093	-0.037	0.089	0.124	0.002	-0.126	-0.071	0.10
0.151	-0.055	-0.100	-0.291	-0.031	0.181	0.126	-0.063	-0.102	0.00
0.223	0.170	-0.014	-0.106	-0.035	0.061	0.052	-0.026	-0.051	-0.00
0.040	0.019	-0.024	-0.025	0.010	0.025	0.002	-0.021	-0.010	0.01
0.014	-0.005	-0.013	-0.001	0.009	0.005	-0.005	-0.007	0.002	0.00
0.001	0.004	-0.002	0.002	0.003	-0.001	0.002	-0.000	0.001	0.00
-0.000	-0.001	0.000	0.001	0.000	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
0.									

SIGNAL TO NOISE RATIO AT INPUT TO RECEIVER = 18.50 DB

SIGNAL TO NOISE POWER OUT = 40.408 DB

SECTION VIII

TRANSCEIVER BREADBOARD

The algorithms mod/demod breadboard is a scale model of a full-duplex digital transceiver. The design philosophy of the breadboard was to scale down the speed of operation of the transceiver and not to curtail the modes of operation or flexibility of the breadboard. This facilitates the testing of any possible transceiver configuration, which might be chosen for the full-scale brassboard.

The speed of operation of the breadboard was chosen to be one one-hundredth of the full scale transceiver. This allows an off-the-shelf commercial mini-processor, the Data General Supernova, to implement the full-duplex transceiver and several testing routines simultaneously. This represents a considerable reduction in cost over designing a special purpose pipeline processor which would be required to implement a full-scale transceiver. In addition, the serial processing, stored program organization of the Supernova and a special signal processing macro-language developed by Philco-Ford facilitates easy modification of the implemented transceiver.

The hundred to one reduction in operating speed implies that all of the signal frequencies associated with the implemented transceiver are reduced by a factor of one hundred. I. e., instead of a 300 to 3000 hertz audio modulation passband, the breadboard has a 3 to 30 hertz modulation passband. Likewise, the breadboard's 80 bit per second digital signal transmission rate represents an 8000 bit per second rate on the full-scale transceiver.

The radio frequency signal generated by the breadboard transmitter and received by the breadboard receiver is centered about a fixed 2000 hertz frequency. This was chosen for several reasons. First, since the digital frequency synthesis techniques, necessary for variable frequency operation, are well known by now, it was not felt that this feature need be demonstrated to prove the feasibility of the digital transceiver. Second, the scaled radio frequency signal was centered in the upper audio frequency region so that conventional narrowband audio channel test and simulation equipment could be used to simulate the radio frequency channel between the transmitter and receiver.

Most of the controls and terminals on the transceiver control panel are self explanatory. The power switch on this panel controls power for all of the equipment associated with the breadboard, and should be the only switch used to shutdown the equipment. The analog modulation and radio

frequency inputs and outputs have 100K ohm and 600 ohm input and output impedances respectively. Their peak unload output and input signal handling capacity is $\pm$ 10 volts. The sampled data output is the same signal as the receiver modulation signal output before it has passed through the final analog low pass filter. This signal has the repetitive spectrum of an unfiltered sampled signal, but does not have any delay or amplitude distortions which are introduced by the analog low pass filter. The digital input, output, and clock ports meet standard 188B specifications, with the exception that the input circuit has no hysteresis. It tends to interrupt an open input circuit as a random sequence of marks and spaces. The rise and fall times of the digital outputs are 25 micro-seconds. The function of the mode control switches will be discussed later.

During normal transceiver operation, the teletype unit is not needed (it may be disconnected from the system) and the controls of the Supernova processor are not used. The Supernova processor should be left in the locked mode, that is the control key removed. However, these facilities may be used to reload or modify the transceiver program.

To reload the transceiver program, the memory dump tape should be placed in the teletype tape reader, the tape reader control switch thrown to "start", and the teletype control switch thrown to "line". The Supernova control key should be inserted and thrown to the "on" position; this activates the processor control panel switches. All data input switches should be thrown down, and the bit 12 switch thrown up. The "reset" and "program load" switches should be pressed in that order to start reading in the program. After reading in the binary loader the paper tape reader will stop. The "continue" switch should be pressed to read in the transceiver program. After the tape is finished reading the transceiver program will automatically restart.

To modify the program, the facilities of the Data General Debug III program are best used. To enter this program, the data switches are set to octal 200, the "stop" and "start" switches are pressed in that order. To restart the transceiver program, location 2 is executed. After modification or reloading the transceiver program, the processor should be relocked.

During switching the transceiver mode of operation between frequency modulation and amplitude modulation or single-sideband modulation, switching transients sometimes give the Supernova processor a larger task than it can perform in real time. Under this condition the transceiver system halts operation. (If the teletype unit is runing it will type out "****ERROR HALT: REALTIME PROGRAM TIMEOUT".) Operation can be most easily restarted by switching the transceiver mode control switches to an easier task, and cycling the power switch "off" and "on". (If the teletype unit is

running it will type out "SYSTEM RESTARTING".)

The possible modes of operation of the breadboard are as follows: In mode A1 no digital signal processing takes place. This mode is used for testing the analog to digital interface and is to be used as a reference when making tests on the digital signal processing. Modes A2 through A4 test the various digital filters used in the transceiver by placing them between the transmitter modulation input and receiver output ports.

Modes B1 through B4 implement frequency modulation operation. Mode B4 implements the complete frequency modulation transceiver. Mode B3 is the same except the 3 to 30 Hz modulation signal bandpass filters are removed. Mode B1 and B2 have the interpolation filters removed also. In mode B1, a phase modulation receiver is implemented in place of the frequency modulation receiver in the other modes. The transmitter still implements frequency modulation.

Modes C1 through C4 implement various amplitude modulations. In modes C1 and C2 amplitude modulation is implemented in the transmitter. In mode C3 upper-sideband and in mode C4 lower-sideband single-sideband modulation is implemented. Upper-sideband reception is implemented in modes C1 and C3, and lower-sideband reception is implemented in modes C2 and C4.

Frequency shift keying digital modulation is implementeed in modes D3 and D4. In mode D3 the digital input is replaced with a pseudo-random sequence generator, and a decoder is placed on the output of the receiver. The signal is so decoded that a continual space output represents error free transmission, and a mark bit represent an error.

The frequency response of various digital filters used in the transceiver were measured and found to have the response shown in figure 44. The 13.5 Hz low pass filter (mode A2) is the filter used to obtain the selectivity in single-sideband reception, and its response represents the selectivity of the digital single-sideband receiver. The bandpass filter (mode A3) and resampling filter (mode A4) responses were measured with them operating at one-half their normal sampling rate. The frequency scale must be doubled to obtain their normal response curve.

Figure 45 shows the response of the frequency modulation transceiver (mode B4) back-to-back. Note that non-complementary pre- and de-emphasis causes a 4db drop in the high frequency response.

Signal-to-noise performance tests were made in all modes of operation. For linear modulation the signal-to-noise performance can be characterize by the apparent noise bandwidth of the receiver. This was found to be

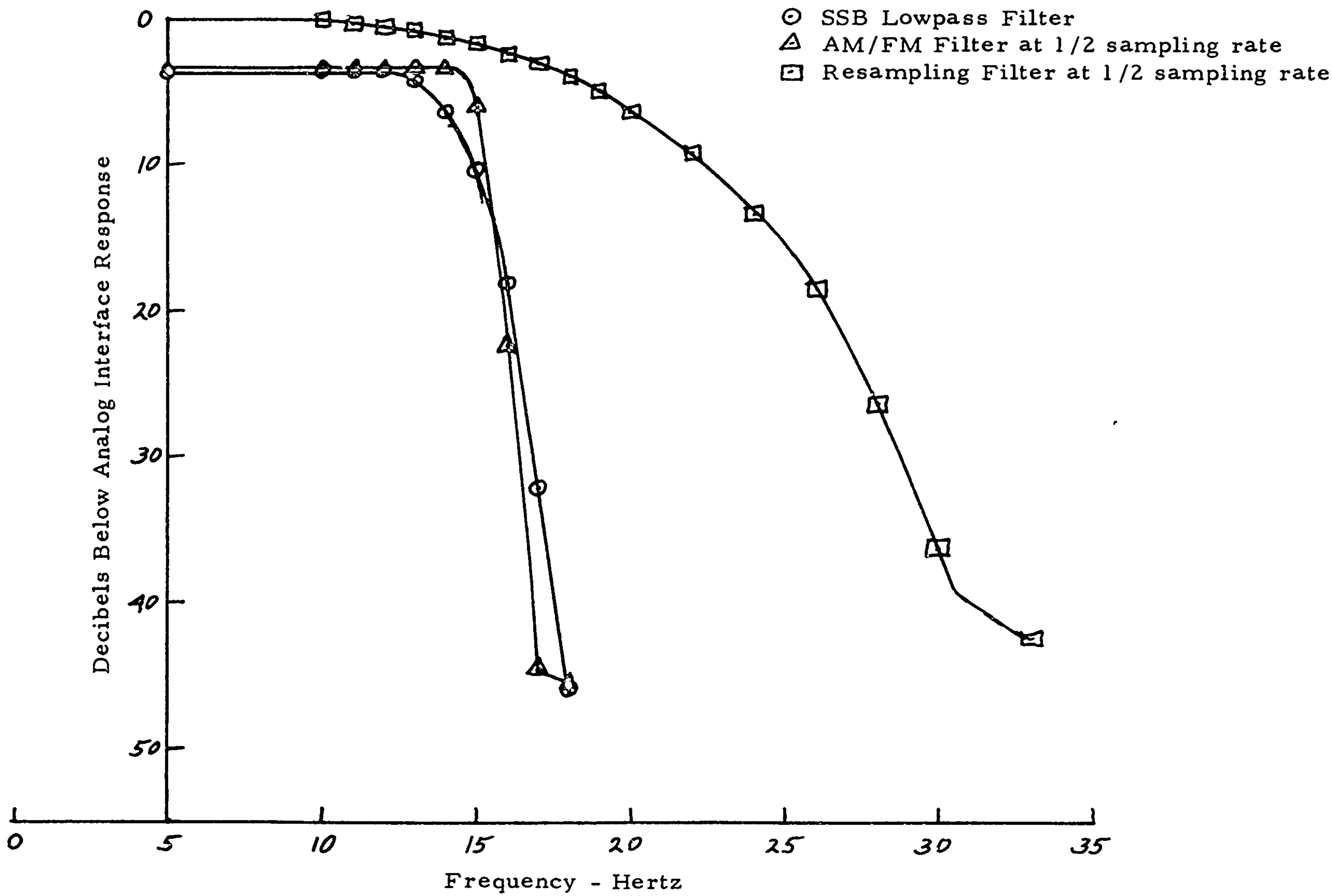


Figure 44: Measured Response of Breadboard Digital Filters

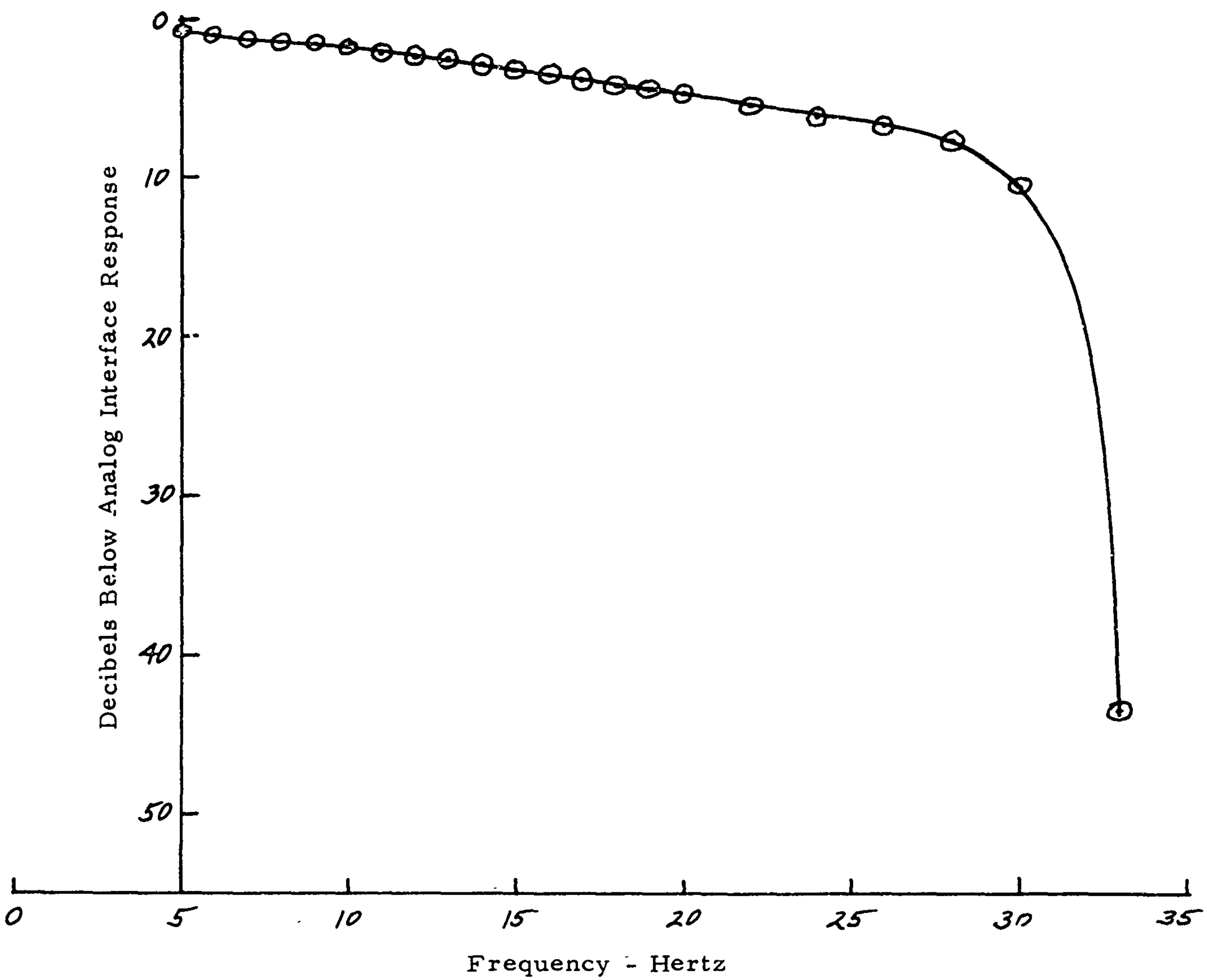


Figure 45: Overall FM Response (Transmitter and Receiver)

approximately 32 Hertz (measured in mode C4). The frequency modulation (mode B4) noise quieting curve is shown in figure 46. The digital transmission error rate versus noise power density curve is shown in figure 47. Note that this measurement was made in mode D3 where the pseudorandom sequence decoder produces a 3 to 1 increase in measured errors above actual errors at low error rates.

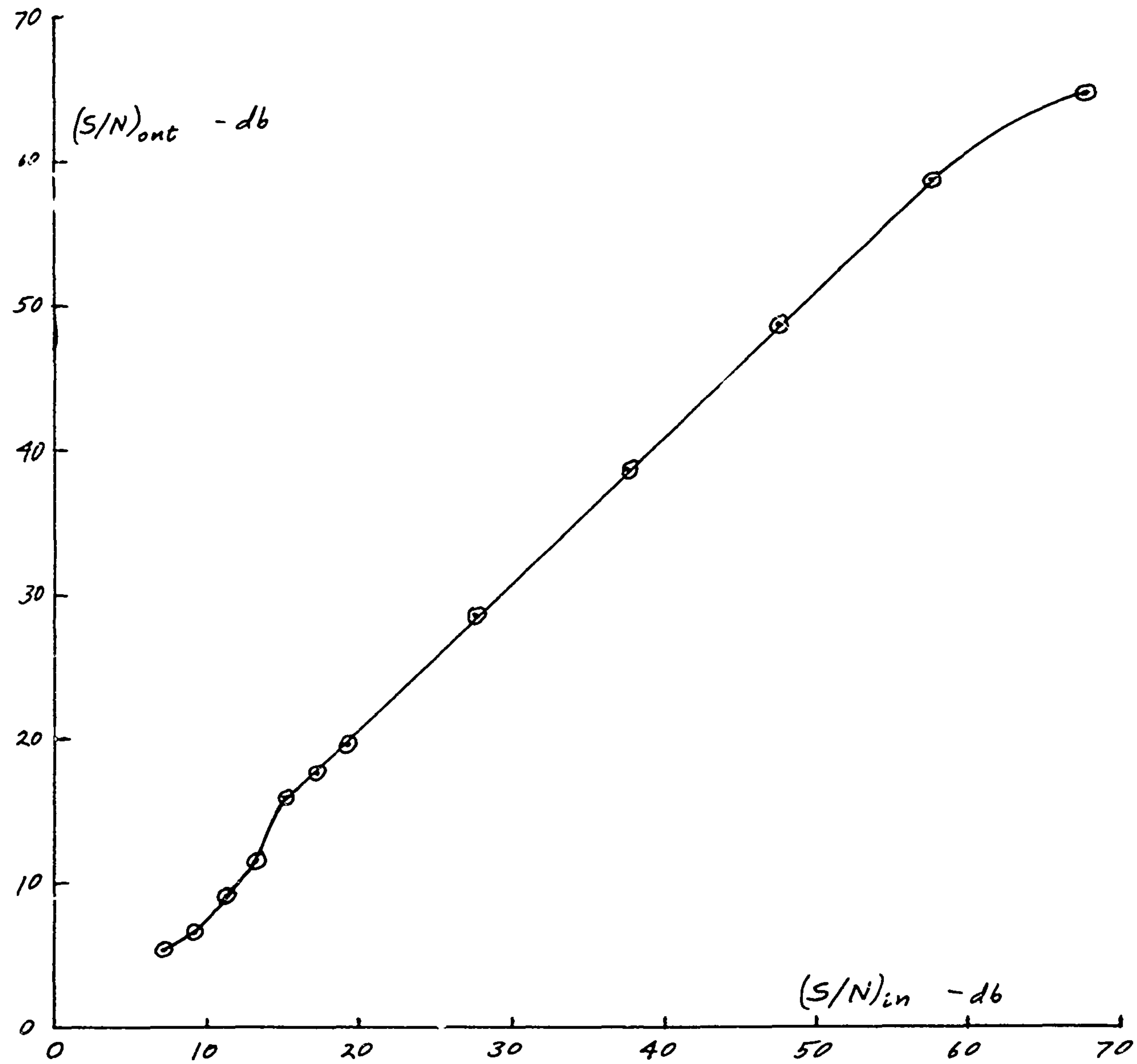


Figure 46: FM Signal-to-Noise Characteristics

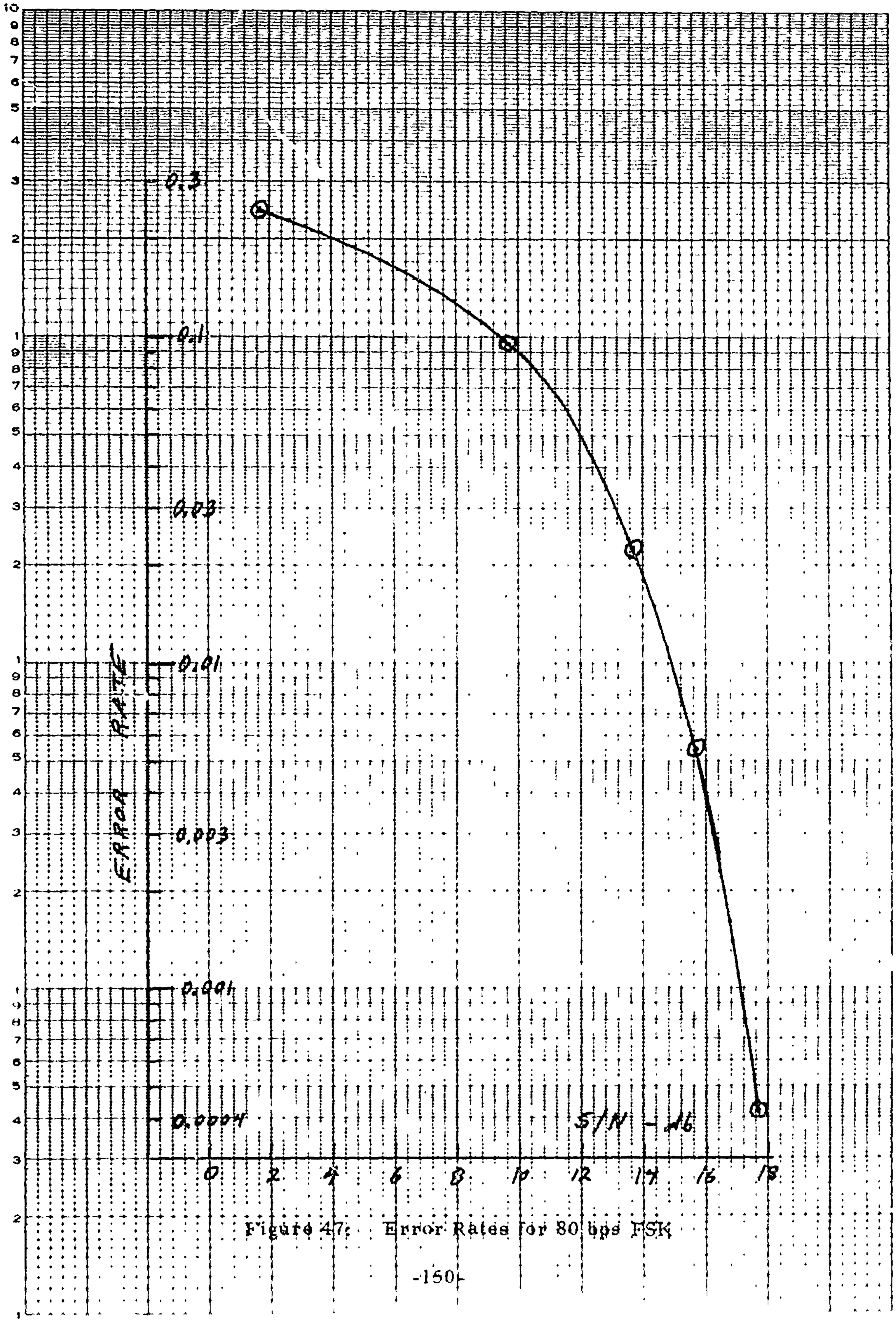


Figure 47: Error Rates for 80 bps FSK

SECTION IX

CONCLUSIONS AND RECOMMENDATIONS

1. CONCLUSIONS

The Digital Transceiver system configuration that took shape as a result of this effort is presented with a minimum of technical detail in the first three subsections of Section II. The system philosophy can be understood with reference to figure 19, in Section V. Filtering of the transmitted signal is performed at baseband. Since the filter requirements can be severe, complexity is minimized by performing this filtering at the lowest possible sampling rate. The resampling filter has less stringent requirements and can operate at a higher output sampling rate. The interpolation filter is extremely simple and can be used to bring the sampling rate up to a value required for translation to I. F. The actual sampling rates required for adequate suppression of spectral repeats are derived in Section II. 4. Section II thus provides a feasible, minimum complexity, overall transceiver system design.

It is readily seen that digital filtering accounts for the major portion of the transceiver. Consequently, a major effort of the program was directed toward efficient filtering algorithms. Section III is a comprehensive survey of all available methods for the design of realizable linear phase digital filters. The method described in Section III. 6 was developed as a rapid technique for designing the many filters required for the breadboard. For the final design of a production item, we would recommend the more cumbersome Hofstetter algorithm described in Section III. 4, since it results in an optimum (minimum complexity) filter design.

A more detailed discussion of digital filtering specifically applied to the digital transceiver, is given in Section VI. The requirements of accuracy, complexity, and linear phase (zero differential delay) lead to the choice of a non-recursive (convolutional) implementation for the sideband filters and a recursive implementation for the interpolation filter. The design of the latter appears in Section II. 4, the design of all the others are given in Section VI. 2 along with frequency responses. Section VI. 2 also contains the interactive filter design computer program.

Since digital differentiation is required for F. M. receiver operation, this topic is discussed in Section VI. 3. A brilliantly efficient method was recently developed by Rabiner and Steiglitz[25] on the basis of a frequency-domain analysis. The time-domain analysis given in Section VI. 3 gives a more penetrating insight into the efficiency of their method. In addition, this analysis provides the required impulse response directly.

The receiver bandpass sampler in figure 19 is meant to extract the complex low-pass equivalent signal by proper sampling of the bandpass received signal. If the narrowband approximation holds; this can be achieved by a pair of analog to digital converters operating a quarter of a carrier cycle apart at a rate consistent with the signal bandwidth. Dickey[19] has recently considered a method for improving the sampler performance for the case when the narrowband approximation is not valid. His method, however, requires the use of three or more A/D convertors. This problem is analyzed in Section IV, and a new method that does not require additional A/D converters, is derived.

Section V may be considered a mathematical appendix for Section II. Extensive computer simulations were performed and are documented in Section VII. These simulations were instrumental in revising the system design to the version discussed in the rest of the report. They proved the feasibility of the digital transceiver, while pointing out that more than eight bit precision need be carried through the various transceiver algorithms.

The transceiver breadboard, described in Section VIII, represents the culmination of our efforts in digital equivalence. By turning switches on the front panel, any of several modes of analog or digital modulation can be implemented. Furthermore, additional modes of modulation and demodulation (analog or digital) can be added to the program as they are invented. A full scale digital transceiver would utilize a special purpose processor to perform (at one hundred times the speed) the same computations that the general purpose processor performs in the breadboard.

Thus, the feasibility of the digital transceiver has been proved and efficient system configurations and processing algorithms have been developed. Advantages of digital processing include stability (nothing can age or drift), size (through LSI), programmability (characteristics may be changed by plugging in a new read-only-memory), commonality (the same processor is an AM transmitter or an FM receiver) and the ability to achieve filter characteristics that are impossible with analog devices.

2. RECOMMENDATIONS

In view of the foregoing conclusions, and of continuing advances in digital integraded circuit technology, it is now feasible to build a real time digital processor for communication signals. It is therefore recommended that a full-scale experimental model Multimode Digital Processing Transceiver be designed, fabricated, and field tested in conjunction with available HF-SSB, VHF-FM, and VHF-AM tactical transceivers.

Two experimental models should be built to be tested in conjunction with existing analog transceivers for voice and CW, and with each other for

digital data transmission via HF and VHF. Digital modulations should include phase shift keying (including differentially coherent PSK), frequency shift keying, and digital data transmission via single sideband (including partial response SSB and carrier injection.

In addition, an L.S.I. investigation should be undertaken so that the potential weight, size, and power advantages of Large Scale Integration may be subsequently realized. This investigation should include:

> Selecting an LSI fabrication technology (MOS, bipolar, etc.) with characteristics capable of implementing system requirements.
>
> Formulating a packaging scheme compatible with the fabrication technology.
>
> Structuring the system such that it is compatible with the fabrication and packaging technology.
>
> Partitioning the logic into LSI blocks to maximize the ratio between the number of logic gates per chip and the number of input/output leads required.
>
> Evaluating the resulting system in terms of cost, size, weight, power, reilability, and maintainability.

It is further recommended that a study be initiated to investigate the possibility of a Universal Signal Processor that would perform vocoding, estimation, channel equalization, synchronization, error detection and correction, and encryption in addition to the Digital Transceiver functions.

REFERENCES

1. B. Fritchman, et. al., "Digital Equivalent Transceiver Study", Philco-Ford report on Contract F30602-68-C-0105, November 1968. RADC-TR-68-539, March 1969, AD 851-364.

2. "Interim Data, Service and Circuit Diagrams, Radio Set AN/PRC-72." Prepared under Contract AF34(601)-23246, October 1967.

3. L. B. W. Jolley, "Summation of Series", 2nd ed., Dover, New York, 1961.

4. R. B. Blackman and J. W. Tukey, "The Measurement of Power Spectra", Dover, New York, 1958.

5. J. F. Kaiser, "Digital Filters", in "System Analysis by Digital Computer" edited by F. F. Kuo and J. F. Kaiser, Wiley, New York, 1966.

6. T. G. Stockham, Jr., "High-Speed Convolution and Correlation with Applications to Digital Filtering", chapter 7 of "Digital Processing of Signals" by B. Gold and C. M. Rader, McGraw-Hill, New York, 1969.

7. H. S. Carslaw, "Introduction to the Theory of Fourier Series and Integrals", 3rd. rev. ed., Dover, New York, 1930.

8. B. Gold and C. M. Rader, "Digital Processing of Signals", McGraw-Hill, New York, 1969, p. 228.

9. H. D. Helms, "Nonrecursive Digital Filters: Design Methods for Achieving Specifications on Frequency Response", IEEE Trans. on Audio and Electroacustics, Vol. AU-16, No. 3, pp. 336-342, September 1968.

10. B. Gold and K. Jordan, "A Direct Search Procedure for Designing Finite Duration Impulse Response Filters", IEEE Trans. on Audio and Electroacoustics, Vol. AU-17, No. 1, pp. 33-36, March 1969.

11. L. R. Rabiner, B. Gold, and C. A. McGonegal, "An Approach to the Approximation Problem for Nonrecursive Digital Filters", IEEE Trans. on Audio and Electroacoustics, Vol. AU-18, No. 2, pp. 83-106, June 1970.

12. L. R. Rabiner, "Techniques for Designing Finite Duration Impulse Response Digital Filters IEEE Trans. on Communication Technology,

12. (Continued)
Vol. Com-19, No. 2, pp. 188-195, April 1971.

13. A. A. G. Requicha and H. B. Voelcker, "Design of Nonrecursive Filters by Specification of Frequency-Domain Zeros", IEEE Trans. on Audio and Electroacoustics, Vol. AU-18, No. 4, pp. 464-470, December 1970.

14. O. Herrman, "Design of Nonrecursive Digital Filters with Linear Phase", Electronics Letters, Vol. 6, No. 11, pp. 328-329, May 28, 1970.

15. E. Hofstetter, A. Oppenheim, and J. Siegel, "A New Technique for the Design of Non-Recursive Digital Filters", to be published. Presented at the Princeton Conference on Information Sciences and Systems, March 1971.

16. E. Averhaus and W. Schüssler, "On the Approximation Problem in the Design of Digital Filters with Limited Wordlength", Archiv der Eleckrischen Übertragung, Band 24, pp. 571-572, 1971.

17. W. Schüssler, "On the Approximation Problem in the Design of Digital Filters". Presented at the Princeton Conference on Information Sciences and Systems, March 1971.

18. R. W. Lucky, J. Salz, and E. J. Weldon, Jr.. "Principles of Data Communication", McGraw-Hill, New York, 1968, pp. 50-51.

19. F. R. Dickey, Jr., "Simple Algorithms for Approximating the Complex Envelope in Terms of Real Samples", Proc. IEEE, Vol. 59, No. 7, pp. 1119-1121, July 1971.

20. K. Abend, et. al., "Advanced Digital Receiver Techniques", Philco-Ford report on Contract F30602-67-C-0178, April 1968. RADC-TR-68-169, September 1968, AD 841-083.

21. T. J. Harley, Jr. and K. Abend, "Digital Noncoherent Amplitude Demodulation and the Modified-Normal Density", Proc. IEEE, Vol. 56, No. 8, pp. 1360-1361, August 1968.

22. S. O. Rice, "Mathematical Analysis of Random Noise", Bell Sys. Tech. J., Vol. 23, pp. 282-332, July 1944, and Vol. 24, pp. 46-156,

22. (Continued)
January 1945. Also in Selected Papers on Noise and Stochastic Processes, N. Wax, Ed., Dover, New York, 1954.

23. E. Christain and E. Eisenmann, "Filter Design Tables and Graphs", John Wiley and Sons, Inc., New York, , p. 243.

24. L. R. Rabiner and R. W. Schafer, "Recursive and Nonrecursive Realizations of Digital Filters Designed by Frequency Sampling Techniques", IEEE Trans. on Audio and Electroacustics, Vol. Au-19, No. 3, pp. 200-207, September 1971.

25. L. R. Rabiner and K. Steiglitz, "The Design of Wide-Band Recursive and Non-Recursive Digital Differentiators", IEEE Trans. on Audio and Electroacoustics, Vol. AU-18, No. 2, pp. 204-209, June 1970.

26. R. Van Blerkom, D. G. Freeman, and R. E. Keeler, IBM report on Contract AF30(602)-3897. RADC-TR-66-602, November 1966, AD 805-009.

27. F. D. Natali and David T. Magill, "Digital Processing Receiver Study", Philco-Ford report on Contract F30602-67-C-0251. RADC-TR-68-163, May 1968, AD 835-767.

APPENDIX A

SUBROUTINES FOR TRANSCEIVER BREADBOARD

```
                .TITL   CONVO
;CONVO IS A SUBROUTINE TO CONVOLVE A 15 BIT 2'S
;COMPLEMENT VECTOR FUNCTION WITH A STATIONARY
;BOUNDED IMPULSE RESPONSE STORED IN EXCESS 2↑15
;FORMAT.  THE CALLING SEQUENCE IS:
;               JSR     @CONVO
;                       GIMP
;                       DSTORE
;WHERE GIMP IS THE STARTING LOCATION OF THE FILTER
;IMPULSE RESPONSE STORAGE BLOCK, AND DSTORE IS THE
;STARTING LOCATION OF THE DATA STORAGE BLOCK ASSOCIATED
;WITH THIS CONVOLUTION.  GIMP AND DSTORE'S FORMAT IS
;AS FOLLOWS:
;GIMP:                  GLENGTH             ;LENGTH OF THE FILTER
;                                           ;IMPULSE RESPONSE
;               .BLK    1                   ;OFFSET CONSTANT=-0.5*
;                                           ;SIGMA(FILTER SAMPLES)
;               .BLK    GLENGTH             ;IMPULSE RESPONSE STORED
;                                           ;IN EXCESS 2↑15 FORMAT
;DSTORE:                .+1                 ;CU[illegible]NT SAMPL[illegible]TER
;               .BLK    GLENGTH             ;DATA VECTOR STORAGE BLK
;THE NEW DATA VECTOR FUNCTION SAMPLE IS TRANSFERRED
;TO THE SUBROUTINE IN AC0.  THE NEW FILTERED DATA VECTOR
;SAMPLE IS RETURNED IN AC0.  AC1, AC2, AC3, AND
;LOCATIONS 20 AND 30 ARE DESTROYED.  CONVO MAY BE
;INTERUPTED IF LOCATIONS 20 AND 30 ARE NOT DESTROYED.
;CONVO DOES NOT REQUIRE INITIALIZATION.  THE ROUTINE
;IS OPTIMIZED FOR SPEED.  SUPERNOVA HARDWARE MULT/DIV
;EXECUTION TIME=26.7+17.0*GLENGTH USEC.S
;TOTAL LENGTH=45. LOCATIONS; 68. FOR THE TOTAL PACKAGE

                .ENT    CONVO, CONVI, CONVE
                .ZREL
00000-000000' CONVO:            ENT1        ;CONVO ENTRY LOCATION
00001-000050' CONVI:            ENT3        ;CONVI ENTRY LOCATION
00002-000064' CONVE:            ENT4        ;CONVE ENTRY LOCATION
                .NREL
00000'054500  ENT1:     STA     3, RETURN   ;SAVE RETURN ADDRESS
00001'103240            ADDOR   0, 0        ;CONVERT DATA SAMPLE
                                            ;TO EXCESS 2↑15 CODE
00002'101620            INCZR   0, 0        ;ROUND & DIVIDE BY 2

00003'031400            LDA     2, 0, 3     ;STARTING LOCATION-1
                                            ;OF IMPULSE RESPONSE
00004'050020            STA     2, INCRE    ;INITIALIZE AUTOINCRE-
                                            ;MENTING IMPULSE
                                            ;RESPONSE SAMPLE POINTER
00005'025000            LDA     1, 0, 2     ;FILTER IMPULSE RESPONSE
                                            ;LENGTH
00006'044473            STA     1, GLENGTH  ;SAVE IN GLENGTH
00007'044473            STA     1, GCOUNT   ;INITIALIZE IMPULSE
                                            ;SAMPLE COUNTER
```

```
00010'035401          LDA     3, 1, 3      ;STARTING LOCATION-1
                                           ;OF DATA BLOCK
00011'031400          LDA     2, 0, 3      ;DATA BLOCK POINTER
00012'041000          STA     0, 0, 2      ;INSERT NEW SAMPLE
00013'151400          INC     2, 2         ;INCREMENT POINTER
00014'050030          STA     2, DECRE     ;SAVE IN AUTODECREMENT-
                                           ;ING DATA BLOCK POINTER
00015'141000          MOV     2, 0         ;INITIALIZE DATA SAMPLE
00016'162400          SUB     3, 0         ;COUNTER
00017'040464          STA     0, DCOUNT
00020'106512          SUBL#   0, 1, SZC    ;IMPLEMENT MODULO
00021'132400          SUB     1, 2         ;GLENGTH
00022'051400          STA     2, 0, 3      ;SAVE NEW DATA BLOCK
                                           ;POINTER

00023'102620 ENT2:    SUBZR   0, 0         ;INITIALIZE ACCUMULATOR
00024'036020          LDA     3, @INCRE    ;TO -IMPULSE RESPONSE
                                           ;OFFSET

00025'014456 LOOP:    DSZ     DCOUNT       ;DECREMENT DATA COUNTER
00026'000406          JMP     CONT.
00027'024452          LDA     1, GLENGTH   ;IF AT DATA BLOCK EDGE -
00030'044453          STA     1, DCOUNT    ;IMPLEMENT MODULO
00031'030030          LDA     2, DECRE     ;GLENGTH OPERATION
00032'133000          ADD     1, 2
00033'050030          STA     2, DECRE
00034'026030 CONT.:   LDA     1, @DECRE    ;DATA SAMPLE
00035'032020          LDA     2, @INCRE    ;IMPULSE RESPONSE SAMPLE
00036'136400          SUB     1, 3         ;COMPENSATE FOR IMPULSE
00037'127000          ADD     1, 1         ;RESPONSE SAMPLE OFFSET
00040'073301          MUL                  ;MULTIPLY
00041'117000          ADD     0, 3         ;ACCUMULATE
00042'121000          MOV     1, 0
00043'014437          DSZ     GCOUNT       ;INCREMENT IMPULSE
                                           ;SAMPLE COUNTER
00044'000761          JMP     LOOP         ;IF NOT FINISHED LOOP

00045'161120          MOVZL   3, 0         ;MOVE 2*RESULT TO AC0
00046'034432          LDA     3, RETURN    ;RETURN
00047'001402          JMP     2, 3

                      ;CONVI IS A SUBROUTINE TO INSERT A NEW SAMPLE IN THE
                      ;DATA STORAGE VECTOR WITHOUT IMPLEMENTING A CONVOLUTION.
                      ;IT IS USED WHEN THE FILTER OUTPUT SAMPLING RATE IS LESS
                      ;THAN THE INPUT SAMPLING RATE.  THE CALLING SEQUENCE IS
                      ;THE SAME AS THAT OF CONVO EXCEPT THE ENTRY POINT IS
                      ;@CONVI.  CONVI REQUIRES NO INITIALIZATION, DESTROYS ALL
                      ;ACCUMULATORS, AND MAY BE INTERUPTED.  SUPERNOVA CORE
                      ;EXECUTION TIME=16.0 USEC.S
                      ;TOTAL LENGTH=13. LOCATIONS

00050'103240 ENT3:    ADDOR   0, 0         ;CONVERT DATA SAMPLE
                                           ;TO EXCESS 2↑15 CODE
00051'101620          INCZR   0, 0         ;DIVIDE BY 2 & ROUND
```

```
00052'033401             LDA     2, @1, 3        ;DATA BLOCK POINTER
00053'041000             STA     0, 0, 2         ;INSERT NEW SAMPLE
00054'151400             INC     2, 2            ;INCREMENT POINTER
00055'023400             LDA     0, @0, 3        ;FILTER IMPULSE RESPONSE
                                                 ;LENGTH
00056'025401             LDA     1, 1, 3         ;STARTING LOCATION-1
                                                 ;OF DATA BLOCK
00057'107000             ADD     0, 1            ;IMPLEMENT MODULO
00060'146512             SUBL#   2, 1, SZC       ;GLENGTH
00061'112400             SUB     0, 2
00062'053401             STA     2, @1, 3        ;SAVE NEW DATA BLOCK
                                                 ;POINTER
00063'001402             JMP     2, 3            ;RETURN

                 ;CONVE IS A SUBROUTINE TO IMPLEMENT A CONVOLUTION
                 ;WITHOUT INSERTING A NEW DATA SAMPLE.  IT IS USED WHEN
                 ;THE FILTER OUTPUT SAMPLING RATE IS GREATER THAN THE
                 ;INPUT SAMPLING RATE.  THE CALLING SEQUENCE IS THE SAME
                 ;AS THAT OF CONVO EXCEPT THE ENTRY POINT IS @CONVE.
                 ;CONVE USES SUBROUTINE CONVO, AND EXCEPT FOR NO DATA
                 ;INPUT IS IDENTICAL TO THAT ROUTINE.  SUPERNOVA HARDWARE
                 ;MULT/DIV EXECUTION TIME=18.7+17.0*GLENGTH USEC.S
                 ;TOTAL LENGTH=9. LOCATIONS

00064'054414  ENT4:      STA     3, RETURN       ;SAVE RETURN ADDRESS
00065'031400             LDA     2, 0, 3         ;STARTING LOCATION-1
                                                 ;OF IMPULSE RESPONSE
00066'050020             STA     2, INCRE        ;INITIALIZE AUTOINCRE-
                                                 ;MENTING IMPULSE
                                                 ;RESPONSE SAMPLE POINTER
00067'025000             LDA     1, 0, 2         ;FILTER IMPULSE RESPONSE
                                                 ;LENGTH
00070'044411             STA     1, GLENGTH      ;SAVE IN GLENGTH
00071'044411             STA     1, GCOUNT       ;INITIALIZE IMPULSE
                                                 ;SAMPLE POINTER
00072'035401             LDA     3, 1, 3         ;STARTING LOCATION-1
                                                 ;OF DATA BLOCK
00073'031400             LDA     2, 0, 3         ;DATA BLOCK POINTER
00074'050030             STA     2, DECRE        ;SET AUTODECRE DATA PTR
00075'172400             SUB     3, 2            ;INITIALIZE DATA SAMPLE
00076'050405             STA     2, DCOUNT       ;COUNTER
00077'000724             JMP     ENT2            ;JUMP TO & FINISH CONVO

       000001 RETURN: .BLK   1               ;RETURN ADDRESS
       000001 GLENGTH:.BLK   1               ;IMPULSE RESPONSE LENGTH
       000001 GCOUNT: .BLK   1               ;IMPULSE SAMPLE COUNTER
       000001 DCOUNT: .BLK   1               ;DATA SAMPLE COUNTER
       000020         .LOC   20
       000001 INCRE:  .BLK   1               ;AUTOINCREMENTING IMP
                                             ;RESPONSE SAMPLE POINTER
       000030         .LOC   30
       000001 DECRE:  .BLK   1               ;AUTODECREMENTING DATA
                                             ;SAMPLE POINTER
                      .END                   ;END OF CONVO, 'VI,& 'VE
```

```
CONT.    000034'
CONVE    000002-
CONVI    000001-
CONVO    000000-
DCOUN    000103'
DECRE    000030
ENT1     000000'
ENT2     000023'
ENT3     000050'
ENT4     000064'
GCOUN    000102'
GLENG    000101'
INCRE    000020
LOOP     000025'
RETUR    000100'
```

```
                  .TITL   COSIN
         ;COSIN IS A SUBROUTINE TO CALCULATE A 16 BIT 2'S
         ;COMPLEMENT SINE AND COSINE FUNCTION FROM A 16 BIT
         ;2*PI'S COMPLEMENT RADIAN ANGLE.  THE ANGLE IS PLACED
         ;IN AC0, RANGE (-PI, PI).  COSINE IS RETURNED IN AC0,
         ;SINE IN AC1, RANGE (-1, 1).  AC2 AND AC3 ARE
         ;DESTROYED.  THE GLOBAL ENTRY POINT IS @COSIN.  THE
         ;ROUTINE IS OPTIMIZED FOR SPEED, MAY BE INTERUPTED,
         ;AND REQUIRES NO INITIALIZATION.  AVERAGE CORE
         ;SUPERNOVA HARDWARE MULT/DIV EXECUTION TIME=64.6 USEC.S
         ;TOTAL LENGTH=125. LOCATIONS

                         .ENT    COSIN
                         .ZREL
00000-000000' COSIN:             ENTRY          ;ENTRY LOCATION
                         .NREL
00000'054464  ENTRY:     STA     3, RETURN      ;SAVE RETURN ADDRESS
00001'040464             STA     0, ANGLE       ;SAVE ANGLE

00002'101300             MOVS    0, 0           ;SHIFT MOST SIGNIFICANT
                                                ;BITS TO RIGHT BYTE
00003'024465             LDA     1, M77         ;SINE TABLE ADDRESS MASK
00004'030465             LDA     2, LSINT       ;SINE TABLE STARTING LOC
00005'115000             MOV     0, 3           ;GENERATE SIN ADDRESS
00006'137400             AND     1, 3                   ;EXTRAT ADDRESS
00007'157000             ADD     2, 3                   ;ADD OFFSET
00010'054456             STA     3, LSIN                ;SAVEIN LSIN
00011'114000             COM     0, 3           ;GENERATE COS ADDRESS
00012'137400             AND     1, 3                   ;EXTRACT ADDRESS
00013'157000             ADD     2, 3                   ;ADD OFFSET
00014'054453             STA     3, LCOS                ;SAVE IN LCOS

00015'101300             MOVS    0, 0           ;RESTORE LOWER ANGLE
                                                ;BITS TO RIGHT BYTE
00016'024454             LDA     1, M377        ;DELTA ANGLE MASK
00017'030454             LDA     2, PIB2        ;2↑16*PI/4
00020'107400             AND     0, 1           ;EXTRACT DELTA ANGLE
00021'102620             SUBZR   0, 0           ;CONVERT TO RADIANS
00022'127120             ADDZL   1, 1                   ;TIMES 4
00023'073301             MUL                            ;TIMES PI/4
00024'030450             LDA     2, PIB8        ;0.5-SIN TABLE OFFSET
00025'113120             ADDZL   0, 2           ;AC2=2*(0.5+SIN(DELTA))

00026'026440             LDA     1, @LSIN       ;COURSE SINE
00027'135220             MOVZR   1, 3           ;FIND -COS(COURSE+DELTA)
00030'102620             SUBZR   0, 0
00031'073301             MUL
```

```
00032'116400          SUB     0, 3           ;AC3=-SIN(DELTA)*SIN(
                                             ;COURSE)
00033'026434          LDA     1, @LCOS       ;COURSE COSINE
00034'137000          ADD     1, 3           ;AC3=COS(COURSE+DELTA)
00035'175112          MOVL#   3, 3, SZC      ;TEST FOR UNITY RESULT
00036'176220          ADCZR   3, 3                  ;IF SO ROUND DWN
00037'054430          STA     3, LCOS        ;SAVE IN LCOS

00040'135220          MOVZR   1, 3           ;FIND SIN(COURSE+DELTA)
00041'102620          SUBZR   0, 0
00042'073301          MUL
00043'162400          SUB     3, 0           ;AC0=SIN(DELTA)*COS(
                                             ;COURSE)
00044'026422          LDA     1, @LSIN       ;COURSE SINE
00045'107000          ADD     0, 1           ;AC1=SIN(COURSE+DELTA)
00046'125112          MOVL#   1, 1, SZC      ;TEST FOR UNITY RESULT
00047'126220          ADCZR   1, 1                  ;IF SO ROUND DWN
00050'020417          LDA     0, LCOS        ;RELOAD COS(COURSE+
                                             ;DELTA)

00051'030414          LDA     2, ANGLE       ;RELOAD ANGLE
00052'151103          MOVL    2, 2, SNC      ;TEST FOR QUADRANTS 3, 4
00053'000403          JMP     .+3
00054'100400          NEG     0, 0                  ;IF SO INVERT
00055'124400          NEG     1, 1                  ;COS, SIN
00056'151103          MOVL    2, 2, SNC      ;TEST FOR EVEN QUADRANTS
00057'000404          JMP     .+4
00060'131000          MOV     1, 2                  ;IF SO -
00061'105000          MOV     0, 1                  ;SIN= COS
00062'140400          NEG     2, 0                  ;COS=-SIN
00063'002401          JMP     @RETURN        ;RETURN


       000001 RETURN: .BLK    1              ;RETURN ADDRES
       000001 ANGLE:  .BLK    1              ;INPUT ANGLE
       000001 LSIN:   .BLK    1              ;SIN ADDRESS
       000001 LCOS:   .BLK    1              ;COS ADDRESS
00070'000077  M77:            77             ;SIN TABLE ADDRESS MASK
00071'000075' LSINT:          SINTAB         ;SINE TABLE LOCATION
00072'000377  M377:           377            ;DELTA ANGLE MASK
00073'144420  PIB2:           51472.         ;2↑16*PI/4
00074'037156  PIB8:           1B1-402.       ;-128*PI IN 0.5'S COMPL

00075'000622  SINTAB:         402.           ;SINTAB IS A TABLE OF
00076'002266                  1206.          ;SINES FROM 0 TO PI/2
00077'003731                  2009.          ;RADIANS.  THE FIRST
00100'005373                  2811.          ;SAMPLE ANGLE=PI/256.
00101'007034                  3612.          ;SAMPLES ARE SPACED
00102'010472                  4410.          ;PI/128. RADIANS APART.
00103'012125                  5205.          ;THE TABLE IS 64.
00104'013556                  5998.          ;SAMPLES LONG.  SINE
00105'015203                  6787.          ;VALUES ARE GIVEN IN
00106'016623                  7571.          ;2'S COMPLEMENT NOTATION
00107'020237                  8351.
```

```
00110'021647          9127.
00111'023250          9896.          ;SINTAB CONTINUED
00112'024644         10660.
00113'026231         11417.
00114'027607         12167.
00115'031156         12910.
00116'032516         13646.
00117'034045         14373.
00120'035363         15091.
00121'036670         15800.
00122'040164         16500.
00123'041446         17190.
00124'042715         17869.
00125'044152         18538.
00126'045373         19195.
00127'046601         19841.
00130'047773         20475.
00131'051151         21097.
00132'052312         21706.
00133'053436         22302.
00134'054544         22884.
00135'055635         23453.
00136'056710         24008.
00137'057744         24548.
00140'060761         25073.
00141'061757         25583.
00142'062736         26078.
00143'063675         26557.
00144'064614         27020.
00145'065513         27467.
00146'066371         27897.
00147'067227         28311.
00150'070043         28707.
00151'070636         29086.
00152'071410         29448.
00153'072140         29792.
00154'072646         30118.
00155'073331         30425.
00156'073773         30715.
00157'074412         30986.
00160'075006         31238.
00161'075357         31471.
00162'075706         31686.
00163'076211         31881.
00164'076472         32058.
00165'076726         32214.
00166'077140         32352.
00167'077326         32470.
00170'077470         32568.
00171'077607         32647.
00172'077702         32706.
00173'077752         32746.
00174'077776         32766.          ;END OF SINTAB
              .END                   ;END OF COSIN
```

```
ANGLE   000065'
COSIN   000000-
ENTRY   000000'
LCOS    000067'
LSIN    C0066'
LSINI   000071'
M377    000072'
M77     000070'
PIB2    000073'
PIB8    000074'
RETUR   000064'
SINTA   000075'
```

```
                    .TITL   ARCTN
                ;ARCTN IS A SUBROUTINE TO CALCULATE A 16 BIT
                ;2*PI'S COMPLEMENT RADIAN ANGLE FROM A 16 BIT
                ;2'S COMPLEMENT COMPLEX VECTOR.  THE REAL PART
                ;OF THE VECTOR IS PLACED IN AC0, THE IMAGINARY
                ;PART IN AC1, RANGLE (-1, 1).  THE ANGLE IS RETURNED
                ;IN AC0, RANGE (-PI, PI).  AC2 AND AC3 ARE DESTROYED.
                ;THE GLOBAL ENTRY POINT IS 0ARCTN.  THE ROUTINE IS
                ;OPTIMIZED FOR SPEED, MAY BE INTERUPTED, AND REQUIRES
                ;NO INITIALIZATION.  AVERAGE CORE SUPERNOVA
                ;HARDWARE MULT/DIV EXECUTION TIME=51.0 USEC.S
                ;TOTAL LENGTH=112. LOCATIONS.

                          .ENT    ARCTN
                          .ZREL
00000-000000' ARCTN:              ENTRY            ;ENTRY LOCATION
                          .NREL
00000'054454  ENTRY:      STA     3, RETURN        ;SAVE RETURN ADDRESS

00001'176440              SUBO    3, 3             ;CLEAR AC3, CARRY
00002'125112              MOVL#   1, 1, SZC        ;IF IMAG NEGITIVE
00003'124460              NEGC    1, 1                     ;NEG, COMPL CRRY
00004'177002              ADD     3, 3, SZC        ;SL1 AC3, TEST CARRY
00005'175400              INC     3, 3                     ;IF CRRY INC AC3
00006'101112              MOVL#   0, 0, SZC        ;IF REAL NEGITIVE
00007'100460              NEGC    0, 0                     ;NEG, COMPL CRRY
00010'177002              ADD     3, 3, SZC        ;SL1 AC3, TEST CARRY
00011'175400              INC     3, 3                     ;IF CRRY INC AC3
00012'106512              SUBL#   0, 1, SZC        ;IF IMAG .GE. REAL
00013'111001              MOV     0, 2, SKP
00014'131061              MOVC    1, 2, SKP                ;SWITCH REAL AND
                                                           ;IMAG, CMPL CRRY
00015'121000              MOV     1, 0
00016'175100              MOVL    3, 3             ;SHIFT CARRY INTO AC3
00017'054436              STA     3, ANGLE         ;SAVE UPPER BITS OF ANGL

00020'145220              MOVZR   2, 1             ;IMAG/REAL
00021'073101              DIV
00022'125002              MOV     1, 1, SZC        ;CHECK FOR OVERFLOW
00023'126060              ADCC    1, 1                     ;IF SO ROUND DWN
00024'121000              MOV     1, 0             ;AC0=AC1=QUOTIENT

00025'030431              LDA     2, M37B4         ;UPPER 5 BIT MASK
00026'034431              LDA     3, LTAB          ;ARCTAN TABLE STARTING
                                                   ;LOCATION
00027'147400              AND     2, 1             ;MASK ADDRESS BITS
00030'125300              MOVS    1, 1             ;SHIFT RIGHT 10.
00031'125220              MOVZR   1, 1
00032'125220              MOVZR   1, 1
00033'137000              ADD     1, 3             ;ADD ARCTAN TABLE
                                                   ;STARTING LOCATION
```

```
00034'025401          LDA     1, 1, 3       ;INTERPOLATING SLOPE
00035'035400          LDA     3, 0, 3       ;COURSE ANGLE
00036'150000          COM     2, 2          ;LOWER 11. BIT MASK
00037'113400          AND     0, 2          ;MASK DELTA ANGLE
00040'102620          SUBZR   0, 0          ;MULTIPLY BY INTERPOL-
                                            ;ATION SLOPE
00041'073301          MUL
00042'117000          ADD     0, 3          ;ADD TO COURSE ANGLE

00043'020412          LDA     0, ANGLE      ;RELOAD UPPER ANGLE BITS
00044'101222          MOVZR   0, 0, SZC     ;SUBTRACT LOWER ANGLE?
00045'174001          COM     3, 3, SKP             ;IF SO NEG LOWER
00046'101221          MOVZR   0, 0, SKP
00047'101620          INCZR   0, 0                  ;ANGLE, BORROW
                                                    ;FRM UPPER ANGLE
00050'101200          MOVR    0, 0
00051'101200          MOVR    0, 0          ;UPPER ANGLE BITS
                                            ;LEFT JUSTIFIED
00052'163000          ADD     3, 0          ;ADD UPPER & LOWER BITS
00053'002401          JMP     @RETURN       ;RETURN


      000001  RETURN: .BLK    1             ;RETURN ADDRESS
      000001  ANGLE:  .BLK    1             ;UPPER ANGLE BITS
00056'174000  M37B4:          37B4          ;UPPER 5 BITS MASK
00057'000060' LTAB:           ARCTAN        ;ARCTAN TABLE LOCATION

00060'000000  ARCTAN:         0.            ;ARCTAN IS A TABLE OF
00061'024273                  10427.        ;ARCTANGIENTS AND
00062'000506                  326.          ;INTERPOLATING SLOPES
00063'024247                  10407.        ;BETWEEN THESE POINTS.
00064'001213                  651.          ;THESE NUMBERS ARE
00065'024176                  10366.        ;LISTED IN PAIRS, THE
00066'001717                  975.          ;ANGLE COMING FIRST.
00067'024102                  10306.        ;THE FIRST PAIR IS FOR
00070'002421                  1297.         ;TANGIENT=0, THE REMAIN-
00071'023763                  10227.        ;ING PAIRS REPRESENT
00072'003121                  1617.         ;TANGIENTS WITH SPACINGS
00073'023622                  10130.        ;OF 1/32. UP TO AND
00074'003615                  1933.         ;INCLUDING TANGIENT=
00075'023440                  10016.        ;31./32.  THE TABLE IS
00076'004306                  2246.         ;32. PAIRS LONG.  THE
00077'023237                  9887.         ;ANGLE IS GIVEN IN
00100'004773                  2555.         ;2*PI'S COMPLEMENT
00101'023016                  9742.         ;RADIANS.  THE SLOPE IN
00102'005454                  2860.         ;PI'S COMPLEMENT
00103'022561                  9585.         ;RADIANS.
00104'006127                  3159.
00105'022310                  9416.
00106'006575                  3453.
00107'022025                  9237.
00110'007236                  3742.
00111'021531                  9049.
00112'007671                  4025.
```

```
00113'021226          8854.          ;ARCTAN CONTINUED
00114'010316          4302.
00115'020715          8653.
00116'010734          4572.
00117'020400          8448.
00120'011344          4836.
00121'020060          8240.
00122'011746          5094.
00123'017535          8029.
00124'012340          5344.
00125'017212          7818.
00126'012725          5589.
00127'016666          7606.
00130'013302          5826.
00131'016343          7395.
00132'013652          6058.
00133'016022          7186.
00134'014212          6282.
00135'015504          6980.
00136'014544          6500.
00137'015170          6776.
00140'015070          6712.
00141'014660          6576.
00142'015405          6917.
00143'014354          6380.
00144'015715          7117.
00145'014053          6187.
00146'016216          7310.
00147'013560          6000.
00150'016512          7498.
00151'013271          5817.
00152'016777          7679.
00153'013007          5639.
00154'017260          7856.
00155'012532          5466.
00156'017532          8026.
00157'012262          5298.          ;END OF ARCTAN
                .END                 ;END OF ARCTN
```

```
---
ANGLE    000055'
ARCTA    000060'
ARCTN    000000-
ENTRY    000000'
LTAB     000057'
M37B4    000056'
RETUR    000054'
```

```
                .TITL   TODRV
            ;TODRV IS A TELETYPE OUTPUT DRIVER SUBROUTINE FOR USE
            ;WITH HARDWARE INTERUPT AND BUFFERED MESSAGE TRANSFERAL.
            ;THE ROUTINE ASSUMES THAT IT HAS BEEN CALLED BY A TTO
            ;INITIATED INTERUPT.  IT PRINTS THE MESSAGE GIVEN IN
            ;THE BUFFER WHOSE STARTING ADDRESS IS LISTED AT TOBUF,
            ;ASSUMING THE BUFFER IS OPEN.  AFTER FINISHING PRINTING
            ;THE BUFFER CONTENSE, THE BUFFER IS CLOSED, TTO IDLED,
            ;AND TTI STARTED.  THE BUFFER STORAGE BLOCK FORMAT IS
            ;AS FOLLOWS:
            ;BUFFER:        BYTEPOINTER
            ;       .TXTE   *MESSAGE*
            ;IF BYTEPOINTER=0, THE BUFFER IS CLOSED,  IF BYTEPOINTER
            ;=1, THE BUFFER IS OPEN AND READY TO PRINT.  THE GLOBAL
            ;ENTRY POINT IS @TODRV.  TTO INTERUPT MUST BE ENABLED
            ;TO INITIATE PRINTING, BUT THIS IS THE ONLY INITIALIZA-
            ;TION REQUIRED BY TODRV.  AVERAGE SUPERNOVA CORE
            ;EXECUTION TIME=19.3 USEC.S
            ;TOTAL LENGTH=17. LOCATIONS

                        .ENT    TODRV, TOBUF
                        .ZREL
00000-000000' TODRV:            ENTRY           ;ENTRY LOCATION
00001-000017' TOBUF:            NULLBUF         ;ADDRESS OF PRINT BUFFER
                        .NREL
00000'030001- ENTRY:    LDA     2, TOBUF        ;BUFFER STARTING LOC
00001'025000            LDA     1, 0, 2         ;BYTE POINTER
00002'125620            INCZR   1, 1            ;INCREMENT ONE
                                                ;EXTRACT WORD ADDRESS

00003'133000            ADD     1, 2            ;ADD OFFSET
00004'021000            LDA     0, 0, 2         ;EXTRACT WORD
00005'125002            MOV     1, 1, SZC       ;TEST FOR UPPER BYTE
00006'101300            MOVS    0, 0                    ;IFSO SWAP BYTES
00007'030411            LDA     2, M377         ;LOWER BYTE MASK
00010'143405            AND     2, 0, SNR       ;MASK CHARACTER
                                                ;TEST FOR NULL CHARACTER
00011'126441            SUBO    1, 1, SKP               ;IF SO CLOSE BUF
00012'061111            DOAS    0, TTO                  ;ELSE PRINT CHAR

00013'125105            MOVL    1, 1, SNR       ;REASSEMBLE BYTE POINTER
                                                ;TEST FOR CLOSED BUFFER
00014'060211            NIOC    TTO                     ;IF SO IDLE TTO
00015'046001-           STA     1, @TOBUF        STORE UPDATED POINTER
00016'001400            JMP     0, 3            ;RETURN

00017'000000  NULLBUF:          0               ;CLOSED BUFFER
00020'000377  M377:             377             ;LOWER BYTE MASK
                        .END                    ;END OF TODRV
```

```
---
ENTRY   000000'
M.77    000020'
NULLB   000017'
TOBUF   000001-
TODRV   000000-
```

```
        .TITL   TIDRV
;TIDRV IS A TELETYPE INPUT DRIVER SUBROUTINE FOR USE
;WITH HARDWARE INTERUPT AND BUFFERED MESSAGE TRANSFERAL.
;THE ROUTINE ASSUMES THAT IT HAS BEEN CALLED BY A TTI
;INITIATED INTERUPT.  IT ECHOS AND READS IN THE MESSAGE
;TYPED ON THE TELETYPE INTO THE BUFFER WHOSE STARTING
;LOCATION IS LISTED AT TIBUF, ASSUMING THE BUFFER IS
;OPEN AND TODRV IS NOT PRINTING SOME OTHER MESSAGE AT
;THE TIME.  A CANCL (CONTRL X) CHARACTER IS ECHOED AS
;A CR, LF, AND CAUSES THE BUFFER TO BE REINITIALIZED, IE
;BYTEPOINTER=1.  A CR CHARACTER IS ECHOED AS A CR, LF,
;AND RECORDED IN THE BUFFER AS AN EOF MARKER. (A NULL
;CHARACTER)  AFTER RECORDING A NULL CHARACTER, THE FILE
;IS CLOSED, AND TTI IDLED.  THE FILE IS AUTOMATICALLY
;CLOSED AFTER 80 CHARACTERS.  THE BUFFER FORMAT IS THE
;SAME AS USED IN TOBUF.  THE GLOBAL ENTRY POINT IS
;@TIDRV.  TTI INTERUPT MUST BE ENABLED TO INITIATE
;THE ROUTINE, BUT NO OTHER INITIATION IS REQUIRED.
;AVERAGE SUPERNOVA CORE EXECUTION TIME=36.8 USEC.S
;TOTAL LENGTH=46. LOCATIONS


                         .ENT    TIDRV, TIBUF
                         .EXTD   MASK, TOBUF
                         .ZREL
00000-000000' TIDRV:             ENTRY              ;ENTRY LOCATION
00001-000055' TIBUF:             NULLBUF            ;ADDRESS OF READ BUFFER
                         .NREL
00000'026002$ ENTRY:     LDA     1, @TOBUF          ;PRINT BUF BYTE POINTER
00001'125004             MOV     1, 1, SZR          ;IS THE PRINT BUF OPEN
00002'000443             JMP     STOP                       ;IF SO STOP READ

00003'026001-            LDA     1, @TIBUF          ;READ BUF BYTE POINTER
00004'125625             INCZR   1, 1, SNR          ;IS THE READ BUF CLOSED
00005'000440             JMP     STOP                       ;IF SO STOP READ
00006'012001-            ISZ     @TIBUF             ;INCREMENT BYTE POINTER

00007'060610             DIAC    0, TTI             ;READ CHAR, CLEAR TTI

00010'030437             LDA     2, CCANCL          ;CANCL (CONTRL X) CHAR
00011'142414             SUB#    2, 0, SZR          ;IS CHARACTER A CANCL
00012'000403             JMP     CRTEST
00013'102520             SUBZL   0, 0                       ;IFSO INITIALIZE
                                                            ;BYTEPOINTER
00014'000421             JMP     CRECHO                     ;AND ECHO CR, LF

00015'030433  CRTEST:    LDA     2, CCR             ;CR CHARACTER
00016'142415             SUB#    2, 0, SNR          ;IS CHARACTER A CR
00017'102460             SUBC    0, 0                       ;IFSO SETTO NULL

00020'030431             LDA     2, C37.            ;37.
00021'146513             SUBL#   2, 1, SNC          ;IS THIS 81TH CHARACTER
00022'102460             SUBC    0, 0                       ;IFSO SETTO NULL
```

```
00023'030001-            LDA    2, TIBUF        ;READ BUF STARTING LOC
00024'133000             ADD    1, 2            ;ADD OFFSET
00025'025000             LDA    1, 0, 2         ;EXTRACT WORD -
00026'125303             MOVS   1, 1, SNC       ;TEST FOR NEW WORD
00027'126460             SUBC   1, 1                    ;IF SO CLEAR
00030'107002             ADD    0, 1, SZC       ;INCERT NEW CHARACTER
                                                ;TEST FOR UPPER BYTE
00031'125300             MOVS   1, 1                    ;IFSO SWAP BYTES
00032'045000             STA    1, 0, 2         ;STORE WORD

00033'101004             MOV    0, 0, SZR       ;TEST FOR NULL CHARACTER
00034'000407             JMP    ECHO
00035'042001-  CRECHO:   STA    0, @TIBUF               ;IF SO CLOSE
00036'102520             SUBZL  0, 0                    ;TIBUF AND ECHO
00037'040414             STA    0, ECHOLF               ;CR, LF
                                                ;OPEN ECHO PRINT BUFFER
00040'020412             LDA    0, LECHO        ;ECHO PRINT BUF LOCATION
00041'040002$            STA    0, TOBUF        ;SAVE AT TOBUF
00042'020406             LDA    0, CCR          ;CR CHARACTER

00043'061111   ECHO:     DOAS   0, TTO          ;START TTO & ECHO CHAR
00044'001400             JMP    0, 3            ;RETURN

00045'060210   STOP:     NIOC   TTI             ;IDLE TTI
00046'001400             JMP    0, 3            ;RETURN


00047'000030   CCANCL:          30              ;CANCL (CONTRL X) CHAR
00050'000215   CCR:             215             ;CR CHARACTER
00051'000045   C37.:            37.             ;37.
00052'000053'  LECHO:           ECHOLF          ;ECHO PRINT BUF LOCATION
00053'000001   ECHOLF:          1               ;ECHO LF PRINT BUFFER
00054'000012                    12              ;LF, NUL CHARACTERS
00055'000000   NULLBUF:         0               ;0 - A CLOSED BUFFER
                         .END                   ;END OF TIDRV
```

```
C37.    000051'
CCANC   000047'
CCR     000050'
CRECH   000035'
CRTES   000015'
ECHO    000043'
ECHOL   000053'
ENTRY   000000'
LECHO   000052'
MASK    000001$X
NULLB   000055'
STOP    000045'
TIBUF   000001-
TIDRV   000000-
TOBUF   000002$X
```

```
                        .TITL   CHCOM
                   ;CHCOM IS A SUBROUTINE PACKAGE FOR BUFFERING THE ANALOG
                   ;INTERFACE UNIT CHANNEL SIGNAL COMMUNICATIONS.  IN
                   ;ADDITION TO CHCOM THE PACKAGE HAS 3 OTHER SUBROUTINES,
                   ;CRDRV, CIDRV, & CODRV, WHICH ARE CALLED DIRECTLY BY AIU
                   ;INITIATED INTERUPTS.  THESE SUBROUTINES IMPLEMENT THE
                   ;CHANNEL COMMUNICATIONS BETWEEN THE AIU AND BUFFER IOBUF
                   ;WHICH HAS THE FOLLOWING FORMAT FOR LINEAR OPERATION:
                   ;IOBUF: .BLK     1               ;FIRST REAL SAMPLE
                   ;       .BLK     1               ;FIRST IMAGINARY SAMPLE
                   ;       .BLK     1               ;SECOND REAL SAMPLE
                   ;       .BLK     1               ;SECOND IMAGINARY SAMLE
                   ;OR FOR ANGLE OPERATION IOBUF HAS THE FOLLOWING FORMAT:
                   ;IOBUF: .BLK     1               ;FIRST ANGLE SAMPLE
                   ;       .BLK     1               ;SECOND ANGLE SAMPLE
                   ;AT THE BEGINING OF A AIU FRAME (DATA SAMPLE PERIOD),
                   ;THE BUFFER CONTAINS TWO SAMPLES OF THE CHANNEL SIGNAL.
                   ;10 SAMPLES WILL BE INTERPOLATED BETWEEN THESE, AND
                   ;TRANSMITTED TO THE CHANNEL.  AT THE SAME TIME 2 SAMPLES
                   ;WILL BE RECEIVED FROM THE CHANNEL, AND WILL BE RECORDED
                   ;IN IOBUF ASSUMING CHANNEL COMMUNICATION IS ENABLED.
                   ;IF CHANNEL COMMUNICATIONS ARE NOT ENABLED, THE BUFFER
                   ;IS NOT DISTURBED.  AT THE BEGINING OF EACH FRAME, CHCOM
                   ;MUST BE CALLED (JSR @CHCOM) TO TRANSFER THROUGH AC0 A
                   ;NEW IOBUF ADDRESS TO THE CHANNEL COMMUNICATION PACKAGE.
                   ;IF AC0=0, CHANNEL COMMUNICATIONS IS DISABLED; IF AC0<>0
                   ;THEN COMM IS ENABLED, AND AC0=2*ADDRESS OF IOBUF+(0 IF
                   ;LINEAR TRANSMISSION, 1 IF ANGLE TRANSMISSION DESIRED).
                   ;THE PACKAGE REQUIRES NO OTHER INITIALIZATION.
                   ;SUPERNOVA CORE EXECUTION TIME=23.3 USEC.S
                   ;TOTAL LENGTH=33. LOCATIONS; 93. FOR THE TOTAL PACKAGE

      000042            .DUSR   AIU=42               ;AIU DEVICE CODE
                        .ENT    CHCOM, CRDRV, CIDRV, CODRV
                        .EXTD   ARCTN, COSIN, MASK
                        .ZREL
00000-000000' CHCOM:            ENT1                 ;INITIALIZATION ENTRY
00001-000025' CRDRV:            ENT2                 ;CH RE INP INTRPT ENTRY
00002-000045' CIDRV:            ENT3                 ;CH IM INP INTRPT ENTRY
00003-000075' CODRV:            ENT4                 ;CH OUTPUT INTRPT ENTRY

                        .NREL
00000'024003$ ENT1:     LDA     1, MASK              ;INTERUPT MASK
00001'030522            LDA     2, MSKMSK            ;AIU, TTI& TTO MASK MASK
00002'147400  CSW:      AND     2, 1                 ;DISABLE CHANNEL COMM
                                                     ;SW=ANGLE INSTRUCTION
00003'101225            MOVZR   0, 0, SNR            ;EXTRACT IOBUF ADDRESS
                                                     ;TST FOR CH COMM REQUEST
00004'000415            JMP     STOP                         ;IF NOT DISABLE
```

```
00005'040521          STA    0, IOBUF        ;SAVE IOBUF ADDRESS
00006'020774          LDA    0, CSW          ;SW=ANGLE INSTRUCTION
00007'101013          MOV#   0, 0, SNC       ;TEST FOR LINEAR OP REQ
00010'101400          INC    0, 0                    ;IF SO MODIFY
                                                     ;SW INSTRUCTION
00011'040471          STA    0, SW1          ;SET SWITCH 1
00012'040436          STA    0, SW2          ;SET SWITCH 2
0001?'020511          LDA    0, TTMASK       ;AIU CH SIGNAL ENABLE, &
                                             ;TTI & TTO DISABLE MASK
00014'040513          STA    0, SAMPLE       ;SYNC SAMPLE COUNTER
00015'107000   FIN:   ADD    0, 1            ;GENERATE NEW MASK
00016'066077          MSKO   1               ;TRANSMIT NEW MASK
00017'044003$         STA    1, MASK         ;SAVE NEW MASK
00020'001400          JMP    0, 3            ;RETURN

00021'062042   STOF:  DOB    0, AIU          ;CLEAR CHANNEL OUTPUT
00022'063042          DOC    0, AIU          ;REGISTERS
00023'020502          LDA    0, TTMASK+1     ;CH SIG DISABLE MASK &
                                             ;TTI & TTO ENABLE MASK
00024'000771          JMP    FIN             ;INITIATE NEW MASK


               ;CRDRV IS AN ANALOG INTERFACE UNIT REAL CHANNEL SIGNAL
               ;INPUT DRIVER SUBROUTINE WHICH IS AN INTIGRAL PART OF
               ;THE CHCOM PACKAGE.  THE ROUTINE ALSO HANDLES REAL DATA
               ;TRANSFER TO AND FROM IOBUF AND COMPUTS THE REAL INCRE-
               ;MENT FOR CODRV.  THE ROUTINE ASSUMES THAT IT HAS BEEN
               ;CALLED BY A CRI (CHANNEL REAL INPUT, 44) INTERUPT.
               ;THE GLOBAL ENTRY POINT IS @CRDRV.  SUPERNOVA HARDWARE
               ;MULT/DIV EXECUTION TIME=26.1 USEC.S
               ;TOTAL LENGTH=17. LOCATIONS

00025'032501   ENT2:  LDA    2, @IOBUF       ;NEW REAL OUTPUT SAMPLE
00026'060542          DIAS   0, AIU          ;INPUT REAL SAMPLE
00027'024473          LDA    1, M17B         ;ANALOG SIGNAL MASK
00030'123400          AND    1, 0            ;MASK OUT CONTROL STATE
00031'042475          STA    0, @IOBUF       ;SAVE REAL INPUT SAMPLE

00032'020476          LDA    0, RFSUM        ;OLD REAL OUTPUT SAMPLE
00033'112400          SUB    0, 2            ;FIND NEW-OLD SAMPLE DIF
00034'102620          SUBZR  0, 0            ;DO SIGNED MULTIPLY
00035'151112          MOVL#  2, 2, SZC               ;TST MULTIPLICAN
00036'150460          NEGC   2, 2                    ;IF NEG, COMPLE
                                                     ;AND SET CARRY
00037'024462          LDA    1, C.1          ;TIMES 0.1
00040'073301          MUL                            ;MULTIPLY
00041'101002          MOV    0, 0, SZC               ;IF CARRY=1
00042'100400          NEG    0, 0                    ;COMPLE RESULT
00043'040466          STA    0, REINC        ;SAVE REAL INCREMENT
00044'001400          JMP    0, 3            ;RETURN
```

```
                ;CIDRV IS AN ANALOG INTERFACE UNIT IMAGINARY CHANNEL
                ;SIGNAL INPUT DRIVER SUBROUTINE WHICH IS AN INTIGAL PART
                ;OF THE CHCOM PACKAGE.  THE ROUTINE ALSO HANDLES
                ;IMAGINARY DATA TRANSFER TO AND FROM IOBUF AND COMPUTS
                ;THE IMAGINARY INCREMENT FOR CODRV.  THE ROUTINE ASSUMES
                ;THAT IT HAS BEEN CALLED BY A CII (CHANNEL IMAGINARY
                ;INPUT, 45) INTERUPT.  SUBROUTINE ARCTN IS USED FOR
                ;POLAR TO LINEAR CONVERSION.  THE GLOBAL ENTRY POINT IS
                ;@CIDRV.  SUPERNOVA HARDWARE MULT/DIV EXECUTION TIME=
                ;31.1 (LINEAR), 18.1+TIME(ARCTN) (ANGLE) USEC.S
                ;TOTAL LENGTH=24. LOCATIONS

00045'054467  ENT3:   STA     3, RETURN       ;SAVE RETURN ADDRESS
00046'064542          DIAS    1, AIU          ;INPUT IMAGINARY SAMPLE
00047'030453          LDA     2, M17B         ;ANALOG SIGNAL MASK
00050'147400  SW2:    AND     2, 1            ;MASK OUT CONTROL STATE
                                              ;LINEAR/ANGLE SWITCH
00051'000417          JMP     PHASE                   ;IF ANGLE GO TO
                                                      ;LIN CONVERSION

00052'010454          ISZ     IOBUF           ;ADVANCE IOBUF POINTER
00053'032453          LDA     2, @IOBUF       ;NEW IMAG OUTPUT SAMPLE
00054'046452          STA     1, @IOBUF       ;SAVE IMAG INPUT SAMPLE

00055'020455          LDA     0, IMSUM        ;OLD IMAG OUTPUT SAMPLE
00056'112400          SUB     0, 2            ;FIND NEW-OLD SAMPLE DIF
00057'102620          SUBZR   0, 0            ;DO SIGNED MULTIPLY
00060'151112          MOVL#   2, 2, SZC               ;IF NEG, COMPLE
00061'150460          NEGC    2, 2                    ;AND SET CARRY
00062'024437          LDA     1, C.1          ;TIMES 0.1
00063'073301          MUL                             ;MULTIPLY
00064'101002          MOV     0, 0, SZC               ;IF CARRY=1
00065'100400          NEG     0, 0                    ;COMPLE RESULT
00066'040445          STA     0, IMINC        ;SAVE IMAG INCREMENT
00067'000404          JMP     END             ;TERMINATE

00070'022436  PHASE:  LDA     0, @IOBUF       ;REAL INPUT SAMPLE
00071'006001$         JSR     @ARCTN          ;CONVERT TO LINEAR
                                              ;NOTATION
00072'042434          STA     0, @IOBUF       ;SAVE INPUT ANGLE SAMPLE

00073'010433  END:    ISZ     IOBUF           ;ADVANCE IOBUF POINTER
00074'002440          JMP     @RETURN         ;RETURN
```

```
;CODRV IS AN ANALOG INTERFACE UNIT CHANNEL SIGNAL OUTPUT
;DRIVER SUBROUTINE WHICH IS AN INTIGRAL PART OF THE
;CHCOM PACKAGE.  THE ROUTINE ASSUMES THAT IT HAS BEEN
;CALLED BY A CSO (CHANNEL SIGNAL OUTPUT, 46) INTERUPT.
;CODRV RETURNS TO AC3+1 ON ODD SAMPLES TO FACILITATE LOW
;SPEED I/O SERVICE WITHOUT INTERFERING WITH AIU SERVICE.
;SUBROUTINE COSIN IS USED FOR LIN TO POLAR CONVERSION.
;THE GLOBAL ENTRY POINT IS @CODRV.  SUPERNOVA CORE
;EXECUTION TIME=28.3 (LINEAR), 24.1+TIME(COSIN) (ANGLE)
;USEC.S.  TOTAL LENGTH=20. LOCATIONS

00075'054437  ENT4:    STA    3, RETURN      ;SAVE RETURN ADDRESS
00076'020432           LDA    0, RESUM       ;CURRENT REAL SAMPLE
00077'030432           LDA    2, REINC       ;REAL SAMPLE INCREMENT
00100'143000           ADD    2, 0           ;REAL INCREMENT
00101'040427           STA    0, RESUM       ;SAVE NEW SAMPLE

00102'147400  SW1:     AND    2, 1           ;LINEAR/ANGLE SWITCH
00103'000406           JMP    ANGLE                  ;IF ANGLE SKIP
                                                     ;IMAG INCREMENT
00104'024426           LDA    1, IMSUM       ;CURRENT IMAG SAMPLE
00105'030426           LDA    2, IMINC       ;IMAG SAMPLE INCREMENT
00106'147000           ADD    2, 1           ;IMAG INCREMENT
00107'044423           STA    1, IMSUM       ;SAVE NEW SAMPLE
00110'000402           JMP    .+2            ;SKIP ANGLE CONVERSION

00111'006002$ ANGLE:   JSR    @COSIN         ;CONVERT TO POLAR
                                             ;NOTATION
00112'062042           DOB    0, AIU         ;OUTPUT REAL SAMPLE
00113'067142           DOCS   1, AIU         ;OUTPUT IMAG SAMPLE
00114'020413           LDA    0, SAMPLE      ;SAMPLE COUNTER
00115'010412           ISZ    SAMPLE         ;INCREMENT SAMPLE COUNT
00116'101212           MOVR#  0, 0, SZC      ;TEST FOR ODD SAMPLE
00117'010415           ISZ    RETURN                 ;IF SO JMP TO
                                                     ;RETURN +1
00120'002414           JMP    @RETURN        ;RETURN


00121'006315  C.1:            3277.          ;2↑16/10
00122'177760  M17B:           177760         ;ANALOG SIGNAL MASK
00123'175774  MSKMSK:         177777-1B5-3   ;AIU, TTI& TTO MASK MASK
00124'000003  TTMASK:         3              ;AIU CH SIGNAL ENABLE, &
                                             ;TTI & TTO DISABLE MASK
00125'002000                  1B5            ;CH SIGNAL DISABLE, &
                                             ;TTI & TTO ENABLE MASK
00126'000130' IOBUF:          RESUM          ;I/O BUFFER POINTER
      000001  SAMPLE: .BLK    1              ;SAMPLE COUNTER
      000001  RESUM:  .BLK    1              ;CURRENT REAL SAMPLE
      000001  REINC:  .BLK    1              ;REAL SAMPLE INCREMENT
      000001  IMSUM:  .BLK    1              ;CURRENT IMAG SAMPLE
      000001  IMINC:  .BLK    1              ;IMAG SAMPLE INCREMENT
      000001  RETURN: .BLK    1              ;RETURN ADDRESS
                      .END                   ;END OF CHCOM PACKAGE
```

ANGLE	000111'
ARCTN	000001$X
CHCOM	000000-
CIDRV	000002-
CODRV	000003-
COSIN	000002$X
CRDRV	000001-
CSW	000002'
C.1	000121'
END	000073'
ENT1	000000'
ENT2	000025'
ENT3	000045'
ENT4	000075'
FIN	000015'
IMINC	000133'
IMSUM	000132'
IOBUF	000126'
M17B	000122'
MASK	000003$X
MSKMS	000123'
PHASE	000070'
REINC	000131'
RESUM	000130'
RETUR	000134'
SAMPL	000127'
STOP	000021'
SW1	000102'
SW2	000050'
TTMAS	000124'

```
                              .TITL   AIDRV
                    ;AIDRV IS A MASTER SUBROUTINE FOR BUFFERING COMMUNICA-
                    ;TIONS WITH THE ANALOG INTERFACE UNIT.  THE ROUTINE
                    ;MAKES USE OF DIDRV TO BUFFER DIGITAL SIGNAL COMMUNICA-
                    ;TION WITH THE AIU, AND THE CHCOM PACKAGE TO BUFFER
                    ;CHANNEL SIGNAL COMMUNICATION.  AIDRV BUFFERS THE LOW
                    ;SPEED (MODULATION) ANALOG SIGNAL COMMUNICATIONS ITSELF.
                    ;THE ROUTINE ASSUMES THAT IT HAS BEEN CALLED BY A ASI
                    ;(ANALOG SIGNAL INPUT, 43) INTERUPT.  THE GLOBAL ENTRY
                    ;POINT IS @AIDRV.  THE ROUTINE NEEDS NO INITIALIZATION.
                    ;ALL COMMUNICATIONS WITH THE REALTIME SIMULATION PROGRAM
                    ;IS THROUGH A BUFFER WHOSE STARTING LOCATOIN IS LISTED
                    ;AT AIBUF.  THE BUFFER FORMAT IS AS FOLLOWS:
                    ;AIBUF:          BUF                ;BUF STARTING LOCATION
                    ;BUF:    .BLK    1                  ;DATA SIGNAL AND CONTROL
                    ;                                   ;BUFFER, BIT 0=DATA BIT
                    ;                                   ;BIT 1=0 LINEAR XMISSION
                    ;                                   ;      =1 ANGLE XMISSION
                    ;                                   ;OVER THE CHANNEL
                    ;                                   ;BIT 2=1; IT IS CLEARED
                    ;                                   ;BEFORE XMISSION OF THE
                    ;                                   ;BUFFERED SIGNALS.
                    ;                                   ;BITS 12-15=CONTRL STATE
                    ;        .BLK    1                  ;ANALOG SIGNAL SAMPLE
                    ;        .BLK    2 OR 4             ;CHANNEL SIGNAL SAMPLES
                    ;                                   ;IN THE FORMAT LISTED IN
                    ;                                   ;THE CHCOM WRITE-UP
                    ;AT THE BEGINING OF A AIU FRAME AIBUF CONTAINS THE
                    ;SIGNAL SAMPLES TO BE OUTPUTED THROUGH THE AIU; AT THE
                    ;END OF THE FRAME, THE BUFFER CONTAINES THE SIGNAL
                    ;SAMPLES INPUTED THROUGH THE AIU.
                    ;SUPERNOVA CORE EXECUTION TIME=33.2 USEC.S
                    ;TOTAL LENGTH=26. LOCATIONS; 33. FOR THE TOTAL PACKAGE

      000042                  .DUSR   AIU=42             ;AIU DEVICE CODE
                              .ENT    AIDRV, AIBUF, DIDRV, CLOCK
                              .EXTD   CHCOM
                              .ZREL
00000-000000' AIDRV:                  ENT1               ;ANALOG INP INTRPT ENTRY
00001-177776  AIBUF:                  -2                 ;AIU COMMUNICATIONS
                                                         ;BUFFER LOCATION
00C02-000027' DIDRV:                  ENT2               ;DATA INPUT INTRPT ENTRY
00003-000000  CLOCK:                  0D                 ;AIU FRAME COUNTER
00004-000000
                              .NREL
00000'024440  ENT1:   LDA     1, DATA                    ;NEW DATA INPUT SAMPLE
00001'054437          STA     3, DATA                    ;SAVE RETURN ADDRESS
00002'074442          DIA     3, AIU                     ;INPUT NEW ANALOG SIGNAL
                                                         ;& CONTROL STATE SAMPLES
00003'020434          LDA     0, M17B                    ;ANALOG SIGNAL MASK
00004'117400          AND     0, 3                       ;MASK
00005'020431          LDA     0, DMASK                   ;DATA, CONTRL STATE MASK
00006'107400          AND     0, 1                       ;MASK
```

```
00007'030001-          LDA     2, AIBUF        ;AIU COMM BUF LOCATION
00010'021000           LDA     0, 0, 2         ;NEW DATA, CONTROL WORD
00011'045000           STA     1, 0, 2         ;SAVE NEW DATA SAMPLE
00012'025001           LDA     1, 1, 2         ;NEW ANALOG SIG SAMPLE
00013'055001           STA     3, 1, 2         ;SAVE NEW ANALOG SAMPLE
00014'125200           MOVR    1, 1            ;APPEND DATA BIT
00015'101100           MOVL    0, 0
00016'125100           MOVL    1, 1
00017'065142           DOAS    1, AIU          ;OUTPUT ANALOG SIGNAL &
                                               ;DATA SAMPLES
00020'151400           INC     2, 2            ;INCRE BUFFER LOCATION
00021'101100           MOVL    0, 0            ;TEST FOR ACTIVE BUFFER
00022'101113           MOVL#   0, 0, SNC       ;COMMUNICATIONS
00023'152040           ADCO    2, 2                    ;IF NOT DISABLE
                                                       ;CHANNEL COMM
00024'141500           INCL    2, 0            ;GENERATE CH COMM IOBUF
                                               ;ADDR & ADD CONTROL BIT
00025'006001$          JSR     @CHCOM          ;INITIALIZE CHANNEL COMM
00026'002412           JMP     @DATA           ;RETURN

                ;DIDRV IS AN ANALOG INTERFACE UNIT DATA SIGNAL INPUT
                ;DRIVER, WHICH IS AN INTIGRAL PART OF THE AIDRV PACKAGE.
                ;THE ROUTINE HANDLES THE DATA AND CONTROL STATE INPUT
                ;TRANSFERS FROM THE AIU. IT ASSUMES THAT IT HAS BEEN
                ;CALLED BY A DSI (DATA SIGNAL INPUT, 42) INTERUPT.  THE
                ;ROUTINE ALSO INCREMENTS AN AIU FRAME COUNTER AND
                ;RETURNS TO AC3+2 TO SIGNAL THE START OF A NEW AIU
                ;SAMPLE FRAME.  THE REALTIME SIMULATION PROGRAM NOW HAS
                ;UNTIL THE ANALOG SIGNAL INTERUPT TO STORE THE NEW AIU
                ;BUFFER ADDRESS IN AIBUF.  THE GLOBAL ENTRY POINT IS
                ;@DIDRV.  THE ROUTINE NEEDS NO INITIALIZATION.
                ;SUPERNOVA EXECUTION TIME=7.1 USEC.S
                ;TOTAL LENGTH=7 LOCATIONS

00027'060542   ENT2:   DIAS    0, AIU          ;INPUT NEW DATA SAMPLE
00030'040410           STA     0, DATA         ;SAVE IN DATA

00031'010004-          ISZ     CLOCK+1         ;INCREMENT AIU FRAME
                                               ;COUNTER
00032'001402           JMP     2, 3            ;RETURN
00033'010003-          ISZ     CLOCK           ;IF CARRY, INCREMENT
                                               ;UPPER COUNTER WORD
00034'001402           JMP     2, 3            ;RETURN
00035'001402           JMP     2, 3


00036'100017   DMASK:          100017          ;DATA, CONTRL STATE MASK
00037'177760   M17B:           177760          ;ANALOG SIGNAL MASK
      000001   DATA:   .BLK    1               ;DATA INPUT STORAGE
                       .END                    ;END OF AIDRV
```

AIBUF	000001-
AIDRV	000000-
CHCOM	000001$X
CLOCK	000003-
DATA	000040'
DIDRV	000002-
DMASK	000036'
ENT1	000000'
ENT2	000027'
M17B	000037'

```
                        .TITL   INTRP
                  ;INTRP IS A MASTER INTERUPT PROGRAM FOR ANSWERING
                  ;HARDWARE INTERUPTS AND PASSING CONTROL OF THE COMPUTER
                  ;ON TO THE APPROPRIATE SOFTWARE DRIVER ROUTINES.  THIS
                  ;PROGRAM ALSO CONTROLS TRANSFER BETWEEN BACKGROUND AND
                  ;REALTIME PROCESSING LEVELS.  INTRP USES THE FOLLOWING
                  ;SUBROUTINES:                      ABEND (ADNORMAL TERMIN)
                  ;START (SYSTEM INITIALIZATION)     RPROG (REALTIME PROGRAM)
                  ;TIDRV (TTI DRIVER)                TODRV (TTO DRIVER)
                  ;CODRV (CH OUT DRIVER)             DIDRV (DATA IN DRIVER)
                  ;AIDRV (ANALOG IN DRIVER)          CRDRV (CH RE IN DRIVER)
                  ;CIDRV (CH IM IN DRIVER)
                  ;INTRP ASSUMES THAT IT HAS BEEN ENTERED THROUGH AN
                  ;HARDWARE INTERUPT.  THE ENTRY POINT IS INTRP.  INTRP
                  ;REQUIRES NO INITIALIZATION.  INTRP USES BIT 0 OF MASK
                  ;AS A REALTIME PROGRAM MASK.  IT ASSUMES THE ONLY
                  ;PERIFERAL DEVICES ARE THE TELETYPE & THE ANALOG INTER-
                  ;FACE UNIT.  SUPERNOVA CORE EXECUTION TIMES ARE:
                  ;       NORMAL INTERUPT EXECUTION TIME  =55.8 USEC.S
                  ;       MULTI-INTERUPT INCREMENTAL TIME =26.7 USEC.S
                  ;       MAXIMUM LATENCY TIME            =50.2 USEC.S
                  ;       INTERUPT TO DRIVER ENTRY DELAY  =28.3 USEC.S
                  ;TOTAL LENGTH=119. LOCATIONS


                        .ENT    INTRP, CMASK, MASK
                        .EXTD   ABEND, START, RPROG, TIDRV, TODRV
                        .EXTD   DIDRV, AIDRV, CRDRV, CIDRV, CODRV

      000000            .LOC    0
00000 002002$           JMP     @START            ;MANUAL START, POWER
                                                  ;DOWN RESTART, OR HARD-
                                                  ;WARE INTERUPT RETURN
                                                  ;ADDRESS
00001 000000'                   INTRP             ;MASTER INTERUPT ROUTINE
                                                  ;ADDRESS
00002 002002$           JMP     @START            ;MANUAL START POINT
00003 000003            JMP     .
00004 000000  CMASK:            0                 ;CURRENT PERIFERALS MASK
                        .ZREL
00000-000000  MASK:             0                 ;CURRENT MASK
```

```
                                .NREL
00000'000401   INTRP:   JMP     .+1             ;INTRP ENTRY POINT
                                                ;PROGRAM LEVEL SWITCH

00001'040554            STA     0, BCS+0        ;SAVE BACKGROUND PROGRAM
                                                ;LEVEL CPU STATE
                                                ;SAVE AC0
00002'044554            STA     1, BCS+1        ;SAVE AC1
00003'050554            STA     2, BCS+2        ;SAVE AC2
00004'054554            STA     3, BCS+3        ;SAVE AC3
00005'020000            LDA     0, 0            ;SAVE CARRY, RETURN ADDR
00006'101100            MOVL    0, 0
00007'040552            STA     0, BCS+4
00010'000411            JMP     PTEST

00011'100002$ RTN:              @START          ;SPARE LOCATION - RETURN
                                                ;ADDRESS

00012'040550            STA     0, RCS+0        ;SAVE REALTIME PROGRAM
                                                ;LEVEL CPU STATE
                                                ;SAVE AC0
00013'044550            STA     1, RCS+1        ;SAVE AC1
00014'050550            STA     2, RCS+2        ;SAVE AC2
00015'054550            STA     3, RCS+3        ;SAVE AC3
00016'020000            LDA     0, 0            ;SAVE CARRY, RETURN ADDR
00017'101100            MOVL    0, 0
00020'040546            STA     0, RCS+4

00021'020473   PTEST:   LDA     0, JSTART       ;RESTORE RESTART INSTR
00022'040000            STA     0, 0            ;TO LOCATION 0
00023'063777            SKPDZ   CPU             ;TEST FOR POWER FAILURE
00024'063077            HALT                            ;IF SO HALT

00025'061477   RTEST:   INTA    0               ;SERVICE PEIFERAL INTRPT
                                                ;ACK INTERUPT
00026 101015            MOV#    0, 0, SNR       ;TEST FOR INTERUPT
00027'000442            JMP     EXIT                    ;IF NONE RETURN
00030'024463            LDA     1, C7           ;DEV CODE TRUNCATION MSK
00031'030464            LDA     2, LINTR        ;DRV ROUTINE TABLE ADDR
00032'107400            AND     0, 1            ;MASK DEVICE CODE
00033'133000            ADD     1, 2            ;GEN DRV ROUNTINE POINTR
00034'007000            JSR     @0, 2           ;EXECUTE DRIVER ROUTINE

00035'000411            JMP     RETURN          ;NORMAL RETURN

00036'000436            JMP     TTEST           ;TELETYPE I/O ENABLED
```

```
00037'004444            JSR     SWITCH          ;NEW AIU FRAME STARTED
                                                ;RESTART REALTIME PROG
                                                ;SW CPU STATE STOR AREAS
00040'101113            MOVL#   0, 0, SNC       ;TEST FOR REALTIME MODE
00041'000440            JMP     TOERR                   ;IF SO ERR HALT
00042'060177            INTEN                   ;ENABLE INTERUPT
00043'006003$           JSR     @RPROG          ;CALL REALTIME PROGRAM
00044'060277            INTDS                   ;DISABLE INTERUPT
00045'004436            JSR     SWITCH          ;SW CPU STATE STOR AREAS

00046'000412  RETURN:   JMP     .+12            ;RESTORE CPU STATE
                                                ;PROGRAM LEVEL SWITCH
00047'020517            LDA     0, RCS+4        ;RESTORE REALTIME PROG
                                                ;LEVEL CPU STATE
                                                ;RESTORE CARRY, RTN ADDR
00050'101220            MOVZR   0, 0
00051'040740            STA     0, RTN          ;SAVE RETURN ADDR IN RTN
00052'034513            LDA     3, RCS+3        ;RESTORE AC3
00053'030511            LDA     2, RCS+2        ;RESTORE AC2
00054'024507            LDA     1, RCS+1        ;RESTORE AC1
00055'020505            LDA     0, RCS+0        ;RESTORE AC0
00056'040476            STA     0, AC0          ;SAVE TEMPORARILY IN AC0
00057'000746            JMP     RTEST           ;TEST FOR MORE INTERUPTS

00060'020501            LDA     0, BCS+4        ;RESTORE BACKGROUND PROG
                                                ;LEVEL CPU STATE
                                                ;RESTORE CARRY, RTN ADDR
00061'101220            MOVZR   0, 0
00062'040727            STA     0, RTN          ;SAVE RETURN ADDR IN RTN
00063'034475            LDA     3, BCS+3        ;RESTORE AC3
00064'030473            LDA     2, BCS+2        ;RESTORE AC2
00065'024471            LDA     1, BCS+1        ;RESTORE AC1
00066'020467            LDA     0, BCS+0        ;RESTORE AC0
00067'040465            STA     0, AC0          ;SAVE TEMPORARILY IN AC0
00070'000735            JMP     RTEST           ;TEST FOR MORE INTERUPTS

00071'020463  EXIT:     LDA     0, AC0          ;RELOAD AC0
00072'060177            INTEN                   ;ENABLE INTERUPT
00073'002716            JMP     @RTN            ;RETURN FOR THE INTERUPT

00074'063710  TTEST:    SKPDZ   TTI             ;TEST TTI
00075'006004$           JSR     @TIDRV                  ;IF DONE SERVICE
00076'063711            SKPDZ   TTO             ;TEST TTO
00077'006005$           JSR     @TODRV                  ;IF DONE SERVICE
00100'000746            JMP     RETURN          ;RETURN

00101'006001$ TOERR:    JSR     @ABEND          ;REALTIME PROGRAM TIME-
00102'000126'                   TOMESS          ;OUT - ERROR TERMINATE
```

```
00103'020675  SWITCH: LDA    0, INTRP       ;SWITCH CPU STATE
00104'024742          LDA    1, RETURN      ;STORAGE AREAS
00105'040741          STA    0, RETURN
00106'044672          STA    1, INTRP
00107'020000-         LDA    0, MASK        ;COMPLEMENT REALTIME
00110'103240          ADDOR  0, 0           ;PROGRAM MASK BIT
00111'040000-         STA    0, MASK        ;SAVE NEW MASK
00112'001400          JMP    0, 3           ;RETURN

00113'000007  C7:            7              ;DEVICE CODE TRUNCATION
                                            ;MASK
00114'002002$ JSTART: JMP    @START         ;MANUAL START OR POWER
                                            ;DWN RESTART INSTRUCTION
00115'000116' LINTR:         INTRL          ;DRV ROUTINE TABLE ADDR
00116'100004$ INTRL:         @TIDRV         ;PERIFERAL DRIVER
                                            ;ROUTINE LOOKUP TABLE
                                            ;TTI (10) DRIVER
00117'100005$                @TODRV         ;TTO (11) DRIVER
00120'100006$                @DIDRV         ;DSI (42) DRIVER
00121'100007$                @AIDRV         ;ASI (43) DRIVER
00122'100010$                @CRDRV         ;CRI (44) DRIVER
00123'100011$                @CIDRV         ;CII (45) DRIVER
00124'100012$                @CODRV         ;CSO (46) DRIVER
00125'100001$                @ABEND         ;NO KNOWN DEVICE CODE

00126'000000  TOMESS:        0              ;REALTIME PROGRAM TIME-
                                            ;OUT ERR MESSAGE
00127'005215          .TXTE  $<215><12>
00130'125012  <12>*
00131'125252  **
00132'142652  *E
00133'151322  RR
00134'151317  OR
00135'044240   H
00136'146101  AL
00137'035324  T:
00140'151240   R
00141'040705  EA
00142'152314  LT
00143'046711  IM
00144'120305  E
00145'151120  PR
00146'043717  OG
00147'152240   T
00150'046711  IM
00151'147705  EO
00152'152125  UT
00153'000000  $

       000001  AC0:   .BLK   1              ;TEMPORARY AC0 STORAGE
       000005  BCS:   .BLK   5              ;BACKGRD CPU STATE STORE
       000005  RCS:   .BLK   5              ;RELTIME CPU STATE STORE
                      .END                  ;END OF INTRP
```

ABEND	000001$X
ACO	000154'
AIDRV	000007$X
BCS	000155'
C7	000113'
CIDRV	000011$X
CMASK	000004
CODRV	000012$X
CRDRV	000010$X
DIDRV	000006$X
EXIT	000071'
INTRL	000116'
INTRP	000000'
JSTAR	000114'
LINTR	000115'
MASK	000000-
PTEST	000021'
RCS	000162'
RETUR	000046'
RPROG	000003$X
RTEST	000025'
RTN	000011'
START	000002$X
SWITC	000103'
TIDRV	000004$X
TODRV	000005$X
TOERR	000101'
TOMES	000126'
TTEST	000074'

```
                              .TITL    START
                      ;START IS AN INITIALIZATION AND BACKGROUND PROGRAM
                      ;INITIATION PROGRAM.  THE GLOBAL ENTRY POINT IS @START.
                      ;START USED SUBROUTINES PRINT AND BPROG.
                      ;TOTAL LENGTH=39. LOCATIONS

      000042                  .DUSR    AIU=42
                              .ENT     START
                              .EXTD    CMASK, MASK, PRINT, BPROG
                              .ZREL
00000-000000' START:                   ENTRY            ;START ENTRY LOCATION

                              .NREL
00000'062677  ENTRY:          IORST                     ;INITIALIZE PERIFERALS
                                                        ;DISABLE INTERUPTS
00001'020001$                 LDA      0, CMASK         ;CURRENT PERIFERALS MASK
00002'040002$                 STA      0, MASK          ;SAVE IN CURRENT MASK

00003'101212                  MOVR#    0, 0, SZC        ;TEST FOR TTY ONLINE
00004'000413                  JMP      SAIU                     ;IF NOT SKIP

00005'126400                  SUB      1, 1             ;WAIT FOR TTY WARM-UP
00006'060011  WAIT:           NIO      TTO
00007'060011                  NIO      TTO
00010'060011                  NIO      TTO
00011'060011                  NIO      TTO
00012'060011                  NIO      TTO
00013'060011                  NIO      TTO
00014'125404                  INC      1, 1, SZR
00015'000771                  JMP      WAIT
00016'065111                  DOAS     1, TTO           ;START TTO

00017'060142  SAIU:           NIOS     AIU              ;START AIU
00020'024411                  LDA      1, C42           ;DSI DEVICE CODE
00021'071477                  INTA     2                ;WAIT FOR BEGINING OF
00022'132414                  SUB#     1, 2, SZR        ;AN AIU SAMPLE FRAME
00023'000776                  JMP      .-2

00024'062177                  DOBS     0, CPU           ;TRANSMIT CURRENT MASK
                                                        ;ENABLE INTERUPTS

00025'006003$                 JSR      @PRINT           ;PRINT STARTUP MESSAGE
00026'000032'                          SMESS            ;STARTUP MESSAGE ADDRESS

00027'006004$                 JSR      @BPROG           ;CALL BACKGROUND
00030'000400                  JMP      .                ;PROGRAM PACKAGE
```

```
00031'000042  C42:               42              ;DSI INTRUPT DEVICE CODE
00032'000000  SMESS:             0               ;STARTUP MESSAGE
00033'005215            .TXTE    $<215><12>
00034'051412  <12>S
00035'051531  YS
00036'142724  TE
00037'120115  M
00040'142722  RE
00041'152123  ST
00042'151101  AR
00043'144724  TI
00044'043516  NG
00045'005215  <215><12>
00046'000012  <12>$

                        END                      ;END OF START
```

```
---
BPROG     000004$X
C42       000031'
CMASK     000001$X
ENTRY     000000'
MASK      000002$X
PRINT     000003$X
SAIU      000017'
SMESS     000032'
START     000000-
WAIT      000006'
```

```
                        .TITL   ABEND
                ;ABEND IS AN ADNORMAL END OF JOB TERMINATION ROUTINE.
                ;IT IS CALLED AS FOLLOWS:
                ;       JSR     @ABEND          ;ERROR TERMINATION
                ;               MESSAGE         ;ERROR PRINTOUT ADDRESS
                ;WHERE THE MESSAGE FORMAT IS AS FOLLOWS:
                ;MESSAGE:       0               ;BYTE POINTER
                ;       .TXTE   $TEXT TO BE PRINTED UPON TERMINATIONS$
                ;THE ERROR MESSAGE NEED NOT BE INCLUDED IT MOST CASES.
                ;AFTER PRINTING THE ERROR MESSAGE, THE LOCATION FROM
                ;WHICH ABEND WAS CALLED+1 IS PRINTED IN OCTAL, AND THE
                ;COMPUTER HALTED.  DURING THIS PERIOD ALL OTHER
                ;INTERUPTS ARE DISABLED.
                ;TOTAL LENGTH=32. LOCATIONS


                        .ENT    ABEND
                        .EXTD   START, PRINT, OCTPR
                        .ZREL
00C00-000000' ABEND:            ENTRY           ;ABEND ENTRY LOCATION

                        .NREL
00000'020420   ENTRY:   LDA     0, ERRMSK       ;ABEND INTERUPT MASK
00001'062177            DOBS    0, CPU          ;TRANSMIT MASK
                                                ;ENABLE INTERUPTS

00002'054436            STA     3, CALL         ;SAVE CALLING LOCATION+1
                                                ;AT CALL
00003'035400            LDA     3, 0, 3
00004'021400            LDA     0, 0, 3         ;ERR MESSAGE BYTEPOINTER
00005'101014            MOV#    0, 0, SZR       ;TEST FOR CLOSED FILE
00006'034413            LDA     3, LERRM                ;IF FILE OPEN
                                                        ;ASSUME ILLEGIT
                                                        ;REPACE WITH
                                                        ;ERRMES

00007'054402            STA     3, .+2          ;PRINT ERROR MESSAGE
00010'006002$           JSR     @PRINT
00011'000022'                   ERRMES

00012'020426            LDA     0, CALL         ;PRINT CALLING LOC+1
00013'006003$           JSR     @OCTPR
00014'006002$           JSR     @PRINT          ;CR - LF
00015'000000                    0

00016'063077            HALT                    ;HALT PROCESSOR
00017'002001$           JMP     @START
```

```
00020'177776  ERRMSK:           177776         ;ABEND INTERUPT MASK
00021'000022' LERRM:            ERRMES         ;DEFAULT PRINTOUT LOC

00022'000000  ERRMES:           0              ;DEFAULT ERROR PRINTOUT
00023'005215          .TXTE     $<215><12>
00024'125012  <12>*
00025'125252  **
00026'040652  *A
00027'142502  BE
00030'042116  ND
00031'040640   A
00032'120324  T
00033'147714  LO
00034'040703  CA
00035'144724  TI
00036'047317  ON
00037'000000  $

      000001  CALL:   .BLK      1              ;CALLING LOC STORAGE
                      .END                     ;END OF ABEND
```

```
---
ABEND   000000-
CALL    000040'
ENTRY   000000'
ERRME   000022'
ERRMS   000020'
LERRM   000021'
OCTPR   000003$X
PRINT   000002$X
START   000001$X
```

```
                .TITL   DECPR
;DECPR AND OCTPR ARE NUMERBICAL PRINTOUT SUBROUTINES.
;DECPR PERFORMS SIGNED DECIMAL CONVERSIONS, AND OCTPR
;UNSIGNED OCTAL CONVERSIONS.  EACH GENERATES AN EIGHT
;PRINTABLE CHARACTER ASCII FIELD WHICH IS PRINTED
;BY SUBROUTINE PRINT.  IT HAS THE FOLOWING FORMAT:
;               DECIMAL:        $ -DDDDD.$
;               OCTAL:          $ DDDDDD $
;NEITHER PROGRAM NEEDS INITIALIZATION, AND BOTH MAY BE
;INTERUPTED.  TOTAL LENGTH=~~56~~ LOCATIONS
                            60.

                .ENT    DECPR, OCTPR
                .EXTD   PRINT
                .ZREL
00000-000006' DECPR:    ENT2            ;DECPR ENTRY LOCATION
00001-000000' OCTPR:    ENT1            ;OCTPR ENTRY LOCATION

                .NREL
00000'054473  ENT1:  STA    3, RETURN   ;SAVE RETURN ADDRESS

00001'034456         LDA    3, CODEL    ;0, DEL CHARACTERS
00002'030451         LDA    2, C10      ;10 OCTAL
00003'105020         MOVZ   0, 1        ;CLEAR CARRY
00004'020452         LDA    0, CSPNUL   ;SP, NULL CHARACTERS
00005'000410         JMP    FILL        ;INITIALIZE PRINT BUFFER

00006'054465  ENT2:  STA    3, RETURN   ;SAVE RETURN ADDRESS

00007'034454         LDA    3, CDELSP   ;DEL, SP CHARACTERS
00010'030444         LDA    2, C10.     ;10 DECIMAL
00011'105020         MOVZ   0, 1        ;CLEAR CARRY
00012'125112         MOVL#  1, 1, SZC   ;TEST FOR NEGITIVE NUM
00013'124460         NEGC   1, 1                ;IFSO NEG,CRRY=1
00014'020441         LDA    0, C.NUL    ;., NULL CHARACTERS

00015'010445  FILL:  ISZ    BUF         ;INITIALIZATION ROUTINE
00016'014444         DSZ    BUF         ;WAIT FOR FREE BUFFER
00017'000776         JMP    .-2

00020'040452         STA    0, BUF+10   ;INITIALIZE PRINT BUFFER
00021'054447         STA    3, BUF+6
00022'054445         STA    3, BUF+5
00023'054443         STA    3, BUF+4
00024'054441         STA    3, BUF+3
00025'054437         STA    3, BUF+2
00026'034431         LDA    3, CODEL    ;0, DEL CHARACTERS
00027'054442         STA    3, BUF+7
```

OK WAW


```
00030'020431            LDA    0, INSTR          ;VARIABLE STORE INSTR
00031'040410            STA    0, STORE          ;INITIALIZE CHAR STORAGE

00032'125015  LOOP:     MOV#   1, 1, SNR         ;NUM TO ASCII CONV LOOP
                                                 ;MOVE QUOTIENT TO NUMER
                                                 ;TEST FOR END CONVERSION
00033'000411            JMP    SIGN                      ;YES JMP SGN TST

00034'175200            MOVR   3, 3              ;SAVE CARRY BIT
00035'102400            SUB    0, 0              ;CLEAR UPPER HALF NUMER
00036'073101            DIV                      ;DIVIDE BY NOTATION BASE
00037'175120            MOVZL  3, 3              ;RESTORE CARRY, AC3

00040'163000            ADD    3, 0              ;ADD REMAINDER TO 0 CHAR
00041'040430  STORE:    STA    0, BUF+7          ;STORE NEW CHAR IN BUF
00042'014777            DSZ    .-1               ;DECREMENT STORAGE LOC
00043'000767            JMP    LOOP              ;LOOP

00044'101062  SIGN:     MOVC   0, 0, SZC         ;TEST FOR NEGITIVE NUM
                                                 ;COMPL NEG NUMBER FLAG
00045'000403            JMP    OUTPUT                    ;IF NOT PRINT

00046'020412            LDA    0, CDELMI         ;DEL, - CHARACTERS
00047'000772            JMP    STORE             ;INCERT MINUS SIGN

00050'006001$ OUTPUT:   JSR    @PRINT            ;PRINT BUFFER CONTENSE
00051'000062'                  BUF
00052'002421            JMP    @RETURN           ;RETURN



00053'000010  C10:             10                ;10 OCTAL
00054'000012  C10.:            10.               ;10 DECIMAL
00055'000056  C.NUL:           56                ;., NULL CHARACTERS
00056'000040  CSPNUL:          40                ;SP, NULL CHARACTERS
00057'177460  CODEL:           377B7+60          ;0, DEL CHARACTERS
00060'026777  CDELMI:          55B7+377          ;DEL, - CHARACTERS

00061'040430  INSTR:    STA    0, BUF+7+.-STORE;VARAIABLE STORE INSTR
                                                 ;BASE VALUE

00062'000000  BUF:             0                 ;PRINT BUFFER BYTEPTR
00063'020377  CDELSP:          40B7+377          ;DEL, SP CHARACTERS
       000007           .BLK   7                 ;BUFFER VARIABLE STORAGE

       000001 RETURN:   .BLK   1                 ;RETURN ADDRESS STORAGE
                        .END                     ;END OF DECPR, OCTPR
```

```
BUF      000062'
CODEL    000057'
C10      000053'
C10.     000054'
CDELM    000060'
CDELS    000063'
CSPNU    000056'
C.NUL    000055'
DECPR    000000-
ENT1     000000'
ENT2     000006'
FILL     000015'
INSTR    000061'
LOOP     000032'
OCTPR    000001-
OUTPU    000050'
PRINT    000001$X
RETUR    000073'
SIGN     000044'
STORE    000041'
```

```
                        .TITL   PRINT
                ;PRINT IS A PRINT SUBROUTINE USED IN CONJUNCTION WITH
                ;THE HARDWARE INTERUPT TTO DRIVER ROUTINE.  THE CALLING
                ;SEQUENCE IS AS FOLLOWS:
                ;       JSR     @PRINT          ;PRINT MESSAGE
                ;               MESSAGE         ;MESSAGE FILE ADDRESS
                ;WHERE THE MESSAGE FORMAT IS AS FOLLOWS:
                ;MESSAGE:       0               ;BYTEPOINTER
                ;       .TXTE   $TEXT TO BE PRINTED$
                ;IF MESSAGE=0, PRINT EXECUTES A CARRIAGE RETURN AND LINE
                ;FEED.  PRINT NEEDS NO INITIALIZATION, AND MAY BE
                ;INTERUPTED.  TOTAL LENGTH=20. LOCATIONS

                        .ENT    PRINT
                        .EXTD   CMASK, TOBUF
                        .ZREL
00000-000000' PRINT:            ENTRY           ;PRINT ENTRY LOCATION

                        .NREL
00000'020001$ ENTRY:    LDA     0, CMASK        ;TEST FOR TELETYPE
00001'101212            MOVR#   0, 0, SZC
00002'001401            JMP     1, 3                    ;IF NONE RETURN

00003'026002$           LDA     1, @TOBUF       ;WAIT FOR TTO IDLE
00004'125004            MOV     1, 1, SZR
00005'000776            JMP     .-2

00006'031400            LDA     2, 0, 3         ;MESSAGE FILE LOCATION
00007'151004            MOV     2, 2, SZR       ;TEST FOR ZERO
00010'000403            JMP     TYPE                    ;IF NOT, PRINT

00011'024407            LDA     1, CCR          ;EXECUTE CR - LF
                                                ;SET FIRST CHARACTER=CR
00012'030407            LDA     2, LLFF         ;SET MESSAGE=LF, NULL

00013'102520  TYPE:     SUBZL   0, 0            ;1
00014'041000            STA     0, 0, 2         ;OPEN MESSAGE FILE
00015'050002$           STA     2, TOBUF        ;SET TTO BUFFER=MESSAGE
                                                ;FILE
00016'065111            DOAS    1, TTO          ;TRANSMIT FIRST CHAR
                                                ;(NOMINALLY A NULL)
                                                ;START TTO
00017'001401            JMP     1, 3            ;RETURN

00020'000215  CCR:              215             ;CR CHARACTER
00021'000022' LLFF:             LFF             ;LOC OF LF, NULL FILE
00022'000000  LFF:              0               ;LF, NULL FILE BYTEPTR
00023'000012                    12              ;LF, NULL
                        .END                    ;END OF PRINT
```

```
---
CCR      000020'
CMASK    000001$X
ENTRY    000000'
LFF      000022'
LLFF     000021'
PRINT    000000-
TOBUF    000002$X
TYPE     000013'
```

```
                 .TITL   MULT
                ;MULT IS A SUBROUTINE TO PERFORM 2'S COMPLEMENT 16 BIT
                ;MULTIPLICATION.  THE CALLING SEQUENCE IS:
                ;        JSR     @MULT
                ;AC0=AC1*AC2; AC1, AC2,& AC3 ARE DESTROYED.
                ;THE ROUTINE MAY BE INTERUPTED.  SUPERNOVA CORE HARDWARE
                ;MULT/DIV EXECUTION TIME, INCLUDING AN INDIRECT SUP-
                ;ROUTINE JUMP=13.1 USEC.S
                ;TOTAL LENGTH=10. LOCATIONS

                         .ENT    MULT
                         .ZREL
00000-000000' MULT:              ENTRY              ;ENTRY LOCATION

                         .NREL
00000'102620  ENTRY:     SUBZR   0, 0               ;GENERATE ROUNDING
00001'101200             MOVR    0, 0               ;CONSTANT
00002'125142             MOVOL   1, 1, SZC          ;SCALE AC1, STORE SIGN,&
                                                    ;TEST FOR NEGITIVE SIGN
00003'124400             NEG     1, 1                     ;IF SO NEGATE
00004'151112             MOVL#   2, 2, SZC          ;IS AC2 NEGITIVE?
00005'150460             NEGC    2, 2                     ;IF SO NEGATE,&
                                                          ;COMPL RESLT SGN
00006'073301             MUL                        ;MULTIPLY
00007'101012             MOV#    0, 0, SZC          ;IS RESULT NEGITIVE?
00010'100400             NEG     0, 0                     ;IF SO NEGATE
00011'001400             JMP     0, 3               ;RETURN

                         .END                       ;END OF MULT
```

ENTRY   000000'
MULT   000000-

```
                         ;MODSW, MDSET                                    PAGE 1 OF 1

                                  .TITL    MODSW
                         ;MDSET IS A SUBROUTINE WHICH ENCODES A 4 BIT CONTROL
                         ;STATE, PASSED TO IT IN BITS 12 THRU 15 OF ACO, INTO A
                         ;16 BIT CONTROL STATE WORD.  THE CONTROL WORD HAS 15
                         ;ZERO BITS AND 1 ONE BIT IN THE BIT POSITION GIVEN BY
                         ;THE BINARY CONTROL STATE INPUT.  THE ENTRY POINT IS
                         ;@MDSET.

                         ;MODSW IS A SWITCH TESTING SUBROUTINE.  IT IS CALLED AS
                         ;FOLLOWS:
                         ;        JSR      @MODSW
                         ;                  MASK
                         ;        RETURN LOCATION FOR NO MATCH
                         ;        RETURN LOCATION FOR MATCH
                         ;A MATCH IS DEFINED AS A ONE BIT IN THE MASK WORD IN
                         ;THE SAME LOCATION AS THE ONE BIT IN THE LAST CONTROL
                         ;STATE WORD GENERATED BY MDSET.  MODSW DOES NOT DESTROY
                         ;ANY ACCUMULATOR OR CARRY BIT.  SUPERNOVA CORE EXECUTION
                         ;TIME, INCLUDING AN INDIRECT SUBROUTINE JUMP, FOR NO
                         ;MATCH=14.0 USEC.S
                         ;TOTAL PACKAGE LENGTH=18. LOCATIONS

                                  .ENT     MODSW, MDSET
                                  .ZREL
00000-000000' MODSW:               ENT1                ;MODSW ENTRY LOCATION
00001-000011' MDSET:               ENT2                ;MDSET ENTRY LOCATION
                                  .NREL
00000'040424  ENT1:      STA       0, ACO              ;SAVE ACO
00001'021400             LDA       0, 0, 3             ;MASK WORD
00002'175400             INC       3, 3                ;GEN NO MATCH RTN ADDR
00003'054422             STA       3, RETURN           ;SAVE AT RETURN
00004'034417             LDA       3, MASK             ;CONTROL WORD
00005'117414             AND#      0, 3, SZR           ;TEST FOR MATCH
00006'010417             ISZ       RETURN                      ;IF SO INCRE RTN
00007'020415             LDA       0, ACO              ;RESTORE ACO
00010'002415             JMP       @RETURN             ;RETURN

00011'024411  ENT2:      LDA       1, M17              ;CONTROL STATE MASK
00012'123400             AND       1, 0                ;MASK
00013'100000             COM       0, 0                ;SET COUNTER
00014'126420             SUBZ      1, 1                ;GENERATE ONE BIT
00015'125200             MOVR      1, 1                ;ADVANCE ONE BIT
00016'101404             INC       0, 0, SZR           ;TEST FOR COPLETE COUNT
00017'000776             JMP       .-2                         ;IF NOT ADVANCE
00020'044403             STA       1, MASK             ;SAVE CONTROL WORD
00021'001400             JMP       0, 3                ;RETURN

00022'000017  M17:                 17                  ;CONTROL STATE MASK
      000001  MASK:      .BLK      1                   ;CONTROL WORD STORAGE
      000001  ACO:       .BLK      1                   ;ACO STORAGE
      000001  RETURN:    .BLK      1                   ;RETURN ADDRESS STORAGE
                         .END                          ;END OF MODSW & MDSET
```

```
---
ACO      000024'
ENT1     000000'
ENT2     000011'
M17      000022'
MASK     000023'
MDSET    000001-
MODSW    000000-
RETUR    000025'
```

```
                    .TITL   LPF1K
                ;LPF1K IS A FILE OF IMPULSE RESPONSE SAMPLES OF A LINEAR
                ;PHASE 1.350 KHZ LOW PASS FILTER.  THE FREQUENCY
                ;RESPONSE IS 1.01 DB DOWN AT 1.350 KHZ, AND GREATER THAN
                ;50.0 DB DOWN ABOVE 1.778 KHZ.  THE SAMPLING RATE IS
                ;8.000 KHZ.  THE SAMPLE VALUES ARE QUANTIZED TO 10 BIT
                ;2'S COMPLEMENT PRECISION.  THE IMPULSE RESPONSE IS 47
                ;SAMPLES LONG.  DC GAIN=0.6665,  THE FORMAT OF THE FILE
                ;IS COMPATABLE WITH THE CONVO SUBROUTINE REQUIREMENTS.

                ;LPFD1 THRU LPFD4 ARE DATA STORAGE FILES FOR USE WITH
                ;LPF1K AND THE CONVO SUBROUTINE.  EACH  1.350 KHZ LOW
                ;PASS FILTER IMPLEMENTED MUST USE A DIFFERENT DATA
                ;STORAGE FILE, BUT MAY SHARE LPF1K AND CONVO FILES.
                ;LPF1K TOTAL LENGTH=49. LOCATIONS.
                ;TOTAL PACKAGE LENGTH=241. LOCATIONS.

                    .ENT    LPF1K, LPFD1, LPFD2, LPFD3, LPFD4
                    .NREL

00000'000057    LPF1K:          47.             ;IMPULSE RESPONSE LENGTH
00001'152530                    -1365.B12       ;OFFSET CONSTANT

00002'100020                    1B0+1.B11       ;IMPULSE RESPONSE
00003'100020                    1B0+1.B11       ;SAMPLSE
00004'077760                    1B0-1.B11
00005'077720                    1B0-3.B11
00006'077760                    1B0-1.B11
00007'100060                    1B0+3.B11
00010'100140                    1B0+6.B11
00011'100000                    1B0+0.B11
00012'077560                    1B0-9.B11
00013'077560                    1B0-9.B11
00014'100140                    1B0+6.B11
00015'100460                    1B0+19.B11
00016'100220                    1B0+9.B11
00017'077300                    1B0-20.B11
00020'077020                    1B0-31.B11
00021'100000                    1B0+0.B11
00022'101340                    1B0+46.B11
00023'101300                    1B0+44.B11
00024'077040                    1B0-30.B11
00025'074660                    1B0-101.B11
00026'076260                    1B0-53.B11
00027'104560                    1B0+151.B11
00030'114360                    1B0+399.B11
00031'117760                    1B0+511.B11
```

```
00032'114360              1B0+399.B11      ;LPF1K CONTINUED
00033'104560              1B0+151.B11
00034'076260              1B0-53.B11
00035'074660              1B0-101.B11
00036'077040              1B0-30.B11
00037'101300              1B0+44.B11
00040'101340              1B0+46.B11
00041'100000              1B0+0.B11
00042'077020              1B0-31.B11
00043'077300              1B0-20.B11
00044'100220              1B0+9.B11
00045'100460              1B0+19.B11
00046'100140              1B0+6.B11
00047'077560              1B0-9.B11
00050'077560              1B0-9.B11
00051'100000              1B0+0.B11
00052'100140              1B0+6.B11
00053'100060              1B0+3.B11
00054'077760              1B0-1.B11
00055'077720              1B0-3.B11
00056'077760              1B0-1.B11
00057'100020              1B0+1.B11
00060'100020              1B0+1.B11        ;END OF LPF1K

00061'000062' LPFD1:      .+1              ;DATA STORAGE FILE 1
      000057        .BLK  47.

00141'000142' LPFD2:      .+1              ;DATA STORAGE FILE 2
      000057        .BLK  47.

00221'000222' LPFD3:      .+1              ;DATA STORAGE FILE 3
      000057        .BLK  47.

00301'000302' LPFD4:      .+1              ;DATA STORAGE FILE 4
      000057        .BLK  47.

                    .END                   ;END OF LPF'1K,' 'D1-'D4
```

LPF1K	000000'
LPFD1	000061'
LPFD2	000141'
LPFD3	000221'
LPFD4	000301'

```
                .TITL   LHRFE
         ;LHRFE AND LHRFO ARE FILES OF THE ODD AND EVEN SAMPLES
         ;RESPECTIVELY OF A 1:2 RESAMPLING FILTER.  THE INPUT
         ;SAMPLING RATE IS 8.000 KHZ, AND THE OUTPUT SAMPLING
         ;RATE IS 16.000 KHZ.  THE FILTER IS LESS THAN 0.01 DB
         ;DOWN AT 1.850 KHZ, AND GREATER THAN 54.9 DB DOWN PAST
         ;8.000-1.350 KHZ.  THE SAMPLES ARE QUANTIZED TO 10 BIT
         ;2'S COMPLEMENT PRECISION.  THE IMPULSE RESPONSE IS
         ;6 SAMPLES (AT THE 8.00 KHZ SAMPLING RATE) LONG.
         ;PASSBAND GAIN=0.9971,  LHRFE AND LHRFO FILE FORMAT IS
         ;COMPATABLE WITH CONVE AND CONVO SUBR REQUIREMENTS.

         ;LHFD1 AND LHFD2 ARE DATA STORAGE FILES FOR USE WITH
         ;LHRFE, LHRFO, AND THE CONVE AND CONVO SUBROUTINES.
         ;EACH RESAMPLING FILTER IMPLEMENTED MUST USE A DIFFERENT
         ;DATA STORAGE FILE, BUT LHRFE, LHRFO, CONVE, AND CONVO
         ;MAY BE SHARED.
         ;LHRFE+LHRFO LENGTH=12. LOCATIONS
         ;TOTAL PACKAGE LENGTH=24. LOCATIONS

                .ENT    LHRFE, LHRFO, LHFD1, LHFD2
                .NREL

00000'000006  LHRFE:           6.               ;IMPULSE RESPONSE LENGTH
00001'140100                   -510.B10         ;OFFSET CONSTANT

00002'100700                   1B0+7.B9         ;ODD IMPULSE RESPONSE
00003'071100                   1B0-55.B9        ;SAMPLES
00004'145700                   1B0+303.B9
00005'145700                   1B0+303.B9
00006'071100                   1B0-55.B9
00007'100700                   1B0+7.B9         ;END OF LHRFE

00010'000006  LHRFO:           6.               ;IMPULSE RESPONSE LENGTH
00011'140040                   -511.B10         ;OFFSET CONSTANT

00012'100000                   1B0+0.B9         ;EVEN IMPULSE RESPONSE
00013'100000                   1B0+0.B9         ;SAMPLES
00014'177700                   1B0+511.B9
00015'100000                   1B0+0.B9
00016'100000                   1B0+0.B9
00017'100000                   1B0+0.B9         ;END OF LHRFO


00020'000021' LHFD1:           .+1              ;DATA STORAGE FILE 1
      000006          .BLK     6.

00027'000030' LHFD2:           .+1              ;DATA STORAGE FILE 2
      000006          .BLK     6.

                      .END                      ;END OF LHRFE PACKAGE
```

LHFD1	000020'
LHFD2	000027'
LHRFE	000000'
LHRFO	000010'

```
                        .TITL   HLRFI
                ;HLRFI IS A FILE OF IMPULSE RESPONSE SAMPLES OF ALINEAR
                ;PHASE 2:1 RESAMPLING FILTER.  THE INPUT SAMPLING RATE
                ;IS 16.000 KHZ, AND THE OUTPUT SAMPLING RATE IS 8.000
                ;KHZ.  THE FILTER IS LESS THAN 0.01 DB DOWN AT 1.350 KHZ
                ;AND GREATER THAN 54.9 DB DOWN PAST 8.000-1.350 KHZ.
                ;THE SAMPLES ARE QUANTIZED TO 10 BIT 2'S COMPLEMENT
                ;PRECISION.  THE IMPULSE RESPONSE IS 11 SAMPLES (AT
                ;THE 16.000 KHZ SAMPLING RATE) LONG.  PASSBAND GAIN=
                ;0.9971,  THE FORMAT OF THE FILE IS COMPATABLE WITH THE
                ;CONVO SUBROUTINE REQUIREMENTS.

                ;HLFD1 AND HLFD2 ARE DATA STORAGE FILES FOR USE WITH
                ;HLRFI AND THE CONVO AND CONVI SUBROUTINES.  EACH
                ;RESAMPLING FILTER IMPLEMENTED MUST USE A DIFFERENT DATA
                ;FILE, BUT MAY SHARE THE HLRFI, CONVO, AND CONVI FILES.
                ;HLRFI LENGTH=11. LOCATIONS.
                ;TOTAL PACKAGE LENGTH=33. LOCATIONS.

                        .ENT    HLRFI, HLFD1, HLFD2
                        .NREL

00000'000013    HLRFI:          11.             ;IMPULSE RESPONSE LENGTH
00001'140060                    -1021.B11       ;OFFSET CONSTANT

00002'100340                    1B0+7.B10       ;IMPULSE RESPONSE
00003'100000                    1B0+0.B10       ;SAMPLES
00004'074440                    1B0-55.B10
00005'100000                    1B0+0.B10
00006'122740                    1B0+303.B10
00007'137740                    1B0+511.B10
00010'122740                    1B0+303.B10
00011'100000                    1B0+0.B10
00012'074440                    1B0-55.B10
00013'100000                    1B0+0.B10
00014'100340                    1B0+7.B10       ;END OF HLRFI

00015'000016'   HLFD1:          .+1             ;DATA STORAGE FILE 1
      000013            .BLK    11.

00031'000032'   HLFD2:          .+1             ;DATA STORAGE FILE 2
      000013            .BLK    11.

                        .END                    ;END OF HLRFI,'FD1,&'FD2
```

```
HLFD1   000015'
HLFD2   000031'
HLRFI   000000'
```

```
                 .TITL   BPF3K
          ;BPF3K IS A FILE OF IMPULSE RESPONSE SAMPLES OF A LINEAR
          ;PHASE 0.300 TO 3.000 KHZ BAND PASS FILTER.  THE
          ;FREQUENCY RESPONSE IS 35.0 DB DOWN AT DC, 1.60 DB DOWN
          ;AT 0.300 AND 3.000 KHZ; AND GREATER THAN 50.0 DB DOWN
          ;ABOVE 3.441 KHZ.  THE SAMPLING RATE IS 16.000 KHZ.  THE
          ;SAMPLE VALUES ARE QUANTIZED TO 10 BIT 2'S COMPLEMENT
          ;PRECISION.  THE IMPULSE RESPONSE IS 107 SAMPLES LONG.
          ;PASSBAND GAIN IS 0.6883,  THE FORMAT OF THE FILE IS
          ;COMPATABLE WITH THE CONVO SUBROUTINE REQUIREMENTS.

          ;BPFD1 AND BPFD2 ARE DATA STORAGE FILES FOR USE WITH
          ;BPF3K AND THE CONVO SUBROUTINE.  EACH 3.0 KHZ BAND PASS
          ;FILTER IMPLEMENTED MUST USE A DIFFERENT DATA STORAGE
          ;FILE, BUT MAY SHARE BPF3K AND CONVO FILES.
          ;BPF3K TOTAL LENGTH=109. LOCATIONS.
          ;TOTAL PACKAGE LENGTH=325. LOCATIONS.

                 .ENT    BPF3K, BPFD1, BPFD2
                 .NREL

00000'000153 BPF3K:      107.           ;IMPULSE RESPONSE LENGTH
00001'177470             -25.B12        ;OFFSET CONSTANT

00002'100020             1B0+1.B11      ;IMPULSE RESPONSE
00003'100000             1B0+0.B11      ;SAMPLES
00004'100000             1B0+0.B11
00005'100000             1B0+0.B11
00006'100020             1B0+1.B11
00007'100020             1B0+1.B11
00010'100020             1B0+1.B11
00011'100000             1B0+0.B11
00012'077760             1B0-1.B11
00013'100000             1B0+0.B11
00014'100040             1B0+2.B11
00015'100040             1B0+2.B11
00016'077760             1B0-1.B11
00017'077720             1B0-3.B11
00020'077760             1B0-1.B11
00021'100040             1B0+2.B11
00022'100040             1B0+2.B11
00023'077740             1B0-2.B11
00024'077640             1B0-6.B11
00025'077660             1B0-5.B11
00026'100020             1B0+1.B11
00027'100040             1B0+2.B11
00030'077700             1B0-4.B11
```

```
00031'077500          1B0-12.B11      ;LPB3K CONTINUED
00032'077500          1B0-12.B11
00033'077720          1B0-3.B11
00034'100020          1B0+1.B11
00035'077620          1B0-7.B11
00036'077300          1B0-20.B11
00037'077240          1B0-22.B11
00040'077560          1B0-9.B11
00041'100000          1B0+0.B11
00042'077560          1B0-9.B11
00043'077060          1B0-29.B11
00044'076720          1B0-35.B11
00045'077340          1B0-18.B11
00046'100000          1B0+0.B11
00047'077560          1B0-9.B11
00050'076640          1B0-38.B11
00051'076300          1B0-52.B11
00052'077040          1B0-30.B11
00053'100060          1B0+3.B11
00054'077760          1B0-1.B11
00055'076440          1B0-46.B11
00056'075440          1B0-78.B11
00057'076400          1B0-48.B11
00060'100400          1B0+16.B11
00061'100720          1B0+29.B11
00062'076320          1B0-51.B11
00063'073400          1B0-144.B11
00064'074520          1B0-107.B11
00065'103340          1B0+110.B11
00066'114040          1B0+386.B11
00067'117760          1B0+511.B11
00070'114040          1B0+386.B11
00071'103340          1B0+110.B11
00072'074520          1B0-107.B11
00073'073400          1B0-144.B11
00074'076320          1B0-51.B11
00075'100720          1B0+29.B11
00076'100400          1B0+16.B11
00077'076400          1B0-48.B11
00100'075440          1B0-78.B11
00101'076440          1B0-46.B11
00102'077760          1B0-1.B11
00103'100060          1B0+3.B11
00104'077040          1B0-30.B11
00105'076300          1B0-52.B11
00106'076640          1B0-38.B11
00107'077560          1B0-9.B11
00110'100000          1B0+0.B11
00111'077340          1B0-18.B11
00112'076720          1B0-35.B11
00113'077060          1B0-29.B11
```

```
00114'077560                    1B0-9.B11      ;BPF3K CONTINUED
00115'100000                    1B0+0.B11
00116'077560                    1B0-9.B11
00117'077240                    1B0-22.B11
00120'077300                    1B0-20.B11
00121'077620                    1B0-7.B11
00122'100020                    1B0+1.B11
00123'077720                    1B0-3.B11
00124'077500                    1B0-12.B11
00125'077500                    1B0-12.B11
00126'077700                    1B0-4.B11
00127'100040                    1B0+2.B11
00130'100020                    1B0+1.B11
00131'077660                    1B0-5.B11
00132'077640                    1B0-6.B11
00133'077740                    1B0-2.B11
00134'100040                    1B0+2.B11
00135'100040                    1B0+2.B11
00136'077760                    1B0-1.B11
00137'077720                    1B0-3.B11
00140'077760                    1B0-1.B11
00141'100040                    1B0+2.B11
00142'100040                    1B0+2.B11
00143'100000                    1B0+0.B11
00144'077760                    1B0-1.B11
00145'100000                    1B0+0.B11
00146'100020                    1B0+1.B11
00147'100020                    1B0+1.B11
00150'100020                    1B0+1.B11
00151'100000                    1B0+0.B11
00152'100000                    1B0+0.B11
00153'100000                    1B0+0.B11
00154'100020                    1B0+1.B11      ;END OF BPF3K

00155'000156' BPFD1:            .+1            ;DATA STORAGE FILE 1
      000153          .BLK      107.

00331'000332' BPFD2:            .+1            ;DATA STORAGE FILE 2
      000153          .BLK      107.

                      .END                     ;END OF BPF3K, 'D1,& 'D2
```

BPF3K	000000'
BPFD1	000155'
BPFD2	000331'

```
                          .TITL   SUBPK

                          .ENT    CONVO, CONVI, CONVE, COSIN, ARCTN
                          .ENT    TODRV, TOBUF, TIDRV, TIBUF, CHCOM
                          .ENT    CRDRV, CIDRV, CODRV, AIDRV, AIBUF
                          .ENT    DIDRV, CLOCK, CMASK, DEAD,  MASK
                          .ENT    START, ABEND, DECPR, OCTPR, PRINT
                          .ENT    MULT,  MODSW, MDSET, RPROG, PPROG

      000004              .LOC    4
00004 000000    CMASK:            0                ;CURRENT PERIFERALS MASK
00005 001400    DEAD:     JMP     0, 3             ;DEAD SUBROUTINE
      000045              .LOC    45
00045 005200    CONVO:            5200             ;CONVOLUTION SUBROUTINE
00046 005250    CONVI:            5200+50          ;ADV INPUT VECTOR ONLY
00047 005264    CONVE:            5200+64          ;ADV OUTPUT VECTOR ONLY
00050 005305    COSIN:            5305             ;COSIN SUBROUTINE ADDR
00051 005503    ARCTN:            5503             ;ARCTAN SUBROUTINE ADDR
00052 005664    TODRV:            5664             ;TTO DRIVER ADDRESS
00053 005703    TOBUF:            5664+17          ;PRINT  BUFFER ADDRESS
00054 005706    TIDRV:            5706             ;TTI DRIVER ADDRESS
00055 005763    TIBUF:            5706+55          ;READ BUFFER ADDRESS
00056 005765    CHCOM:            5765             ;CH COMM SUBROUTINE ADDR
00057 006012    CRDRV:            5765+25          ;CRI DRIVER ADDRESS
00060 006032    CIDRV:            5765+45          ;CII DRIVER ADDRESS
00061 006062    CODRV:            5765+75          ;CSO DRIVER ADDRESS
00062 006124    AIDRV:            6124             ;ASI DRIVER ADDRESS
00063 177776    AIBUF:            -2               ;AIU COMM BUFFER ADDRESS
00064 006153    DIDRV:            6124+27          ;DSI DRIVER ADDRESS
00065 000000    CLOCK:            0D               ;AIU FRAME COUNTER
00066 000000
00067 000000    MASK:             0                ;CURRENT INTERUPT MASK
00070 006355    START:            6355             ;START-UP ROUTINE ADDR
00071 006424    ABEND:            6424             ;ABEND ROUTINE ADDRESS
00072 006473    DECPR:            6465+6           ;DECPR SUBROUTINE ADDR
00073 006465    OCTPR:            6465             ;OCTPR SUBROUTINE ADDR
00074 006562    PRINT:            6562             ;PRINT SUBROUTINE ADDR
00075 006607    MULT:             6607             ;2'S COMPL 16 BIT MULT
00076 006621    MODSW:            6607+12          ;MODSW SUBROUTINE ADDR
00077 006632    MDSET:            6607+12+11       ;MDSET SUBROUTINE ADDR
00100 000005    RPROG:            DEAD             ;REAL TIME PROGRAM ADDR
00101 000005    BPROG:            DEAD             ;BACKGROUND PROGRAM ADDR

                          .END
```

ABEND	000071
AIBUF	000063
AIDRV	000062
ARCTN	000051
BPROG	000101
CHCOM	000056
CIDRV	000060
CLOCK	000065
CMASK	000004
CODRV	000061
CONVE	000047
CONVI	000046
CONVO	000045
COSIN	000050
CRDRV	000057
DEAD	000005
DECPR	000072
DIDRV	000064
MASK	000067
MDSET	000077
MODSW	000076
MULT	000075
OCTPR	000073
PRINT	000074
RPROG	000100
START	000070
TIBUF	000055
TIDRV	000054
TOBUF	000053
TODRV	000052

```
                     .TITL    IMPAC

                     .ENT     LPF1K, LPFD1, LPFD2, LPFD3, LPFD4
                     .ENT     LHRFE, LHRFO, LHFD1, LHFD2, HLRFI
                     .ENT     HLFD1, HLFD2, BPF3K, BPFD1, BPFD2


004000               .LOC     4000
000061    LPF1K:     .BLK     49.              ;1.35 KHZ LPF IMP RESP
000060    LPFD1:     .BLK     48.              ;LPF DATA STORAGE FILE 1
000060    LPFD2:     .BLK     48.              ;LPF DATA STORAGE FILE 2
000060    LPFD3:     .BLK     48.              ;LPF DATA STORAGE FILE 3
000060    LPFD4:     .BLK     48.              ;LPF DATA STORAGE FILE 4
004363               .LOC     4363
000010    LHRFE:     .BLK     8.               ;1:2 RESAMPL FLT IMP RES
000010    LHRFO:     .BLK     8.               ;ODD TERMS OF ABOVE
000007    LHFD1:     .BLK     7.               ;LHRF DATA STORE FILE 1
000007    LHFD2:     .BLK     7.               ;LHRF DATA STORE FILE 2
000015    HLRFI:     .BLK     13.              ;2:1 RESAMPL FLT IMP RES
000014    HLFD1:     .BLK     12.              ;HLRF DATA STORE FILE 1
000014    HLFD2:     .BLK     12.              ;HLRF DATA STORE FILE 2
004470               .LOC     4470
000155    BPF3K:     .BLK     109.             ;0.3-3.0 KHZ BPF IMP RES
000154    BPFD1:     .BLK     108.             ;BPF DATA STORAGE FILE 1
000154    BPFD2:     .BLK     108.             ;BPF DATA STORAGE FILE 2

                     .END
```

BPF3K	004470
BPFD1	004645
BPFD2	005021
HLFD1	004436
HLFD2	004452
HLRFI	004421
LHFD1	004403
LHFD2	004412
LHRFE	004363
LHRFO	004373
LPF1K	004000
LPFD1	004061
LPFD2	004141
LPFD3	004221
LPFD4	004301

APPENDIX B

MAIN PROGRAM FOR TRANSCEIVER BREADBOARD

```
SAFE =
*3
*1
*1
*1
*6
     NMAX 003761
     ZMAX 000050

    ABEND 000071
    AIBUF 000063
    AIDRV 000062
    ARCTN 000051
    BPF3K 004470
    BPFD1 004645
    BPFD2 005021
    BPROG 000101
    CHCOM 000056
    CIDRV 000060
    CLOCK 000065
    CMASK 000004
    CODRV 000061
    CONVE 000047
    CONVI 000046
    CONVO 000045
    COSIN 000050
    CRDRV 000057
    DEAD  000005
    DEBUG 000200
    DECPR 000072
    DIDRV 000064
    HLFD1 004436
    HLFD2 004452
    HLRFI 004421
    LHFD1 004403
    LHFD2 004412
    LHRFE 004363
    LHRFO 004373
    LPF1K 004000
    LPFD1 004061
    LPFD2 004141
    LPFD3 004221
    LPFD4 004301
    MASK  000067
    MDSET 000077
    MODSW 000076
    MULT  000075
    OCTPR 000073
    PRINT 000074
    RPROG 000100
    START 000070
    TIBUF 000055
    TIDRV 000054
    TOBUF 000053
    TODRV 000052
*8
$
44/007552 /000050
7553   007551
7554   007337
7555   003761
7556   177777
```

```
6650    NIOC CPU
6651    LDA 0 +63
6652    LDA 1 120
6653    STA 1 +63
6654    STA 0 120
6655    NIOS CPU

6656    STA 3 121

6657    LDA 0 @120
6660    MOVL 0 1
6661    SUBR 1 1
6662    STA 1 122
6663    JSR @+77

6664    JSR @+76
6665    070000
6666    JMP 6730

6667    LDA 2 120
6670    LDA 1 123
6671    LDA 0 122
6672    ADD 1 0
6673    STA 0 +0 2
6674    LDA 0 +4 2
6675    NEG 0 0
6676    STA 0 +4 2
6677    LDA 0 +5 2
6700    NEG 0 0
6701    STA 0 +5 2
6702    LDA 0 +1 2

6703    JSR @+76
6704    040000
6705    JMP 6712
6706    JSR @+45
6707    004000
6710    004061
6711    JMP 6724

6712    JSR @+76
6713    020000
6714    JMP 6721
6715    JSR @+45
6716    004470
6717    004645
6720    JMP 6724

6721    JSR @+45
6722    004421
6723    004436
6724    NEG 0 0
6725    LDA 2 120
6726    STA 0 +1 2
6727    JMP @121
```

```
6730   JSR @+76
6731   107400
6732   JMP @121
6733   LDA 2 120
6734   LDA 1 124
6735   LDA 0 122
6736   ADD 1 0
6737   STA 0 +0 2

6740   LDA 1 +3 2
6741   ADDOR 1 1
6742   STA 1 +3 2
6743   LDA 0 +1 2
6744   JSR @+76
6745   077777
6746   JMP 6724
6747   LDA 0 +2 2

6750   JSR @+76
6751   004000
6752   JMP 6755
6753   JMP 7010
6754   000000

6755   LDA 3 126
6756   STA 1 126
6757   SUBZL 0 1
6760   STA 1 127
6761   SUBZL 3 0

6762   JSR @+76
6763   001000
6764   JMP 6775
6765   JSR @+46
6766   004421
6767   004436
6770   LDA 0 127
6771   JSR @+45
6772   004421
6773   004436
6774   JMP 7010

6775   JSR @+76
6776   000400
6777   JMP 7010
7000   JSR @+46
7001   004470
7002   004645
7003   LDA 0 127
7004   JSR @+45
7005   004470
7006   004645
7007   ADD 0 0
```

```
7010    LDA 2 120
7011    LDA 1 +1 2
7012    STA 0 +1 2
7013    MOV 1 0

7014    JSR @+76
7015    002000
7016    JMP 7022
7017    MOVL 1 0
7020    MOVR 1 0
7021    JMP 7034

7022    JSR @+76
7023    001000
7024    JMP 7031
7025    JSR @+45
7026    004421
7027    004452
7030    JMP 7034

7031    JSR @+45
7032    004470
7033    005021

7034    LDA 2 120
7035    LDA 3 130
7036    SUB 0 3
7037    STA 3 130
7040    STA 3 +2 2

7041    JSR @+76
7042    001777
7043    JMP 7057
7044    SUB 0 0

7045    JSR @+76
7046    001000
7047    JMP 7054
7050    JSR @+45
7051    004421
7052    004452
7053    JMP 7057

7054    JSR @+45
7055    004470
7056    005021

7057    LDA 2 120
7060    LDA 3 130
7061    SUB 0 3
7062    STA 3 130
7063    STA 3 +3 2
7064    JMP 0121
```

```
$=
 63/007532
100/006650
120/007540
123/020000
124/060000
```

```
185    032314

6732   JMP 7070


7070   JSR @+76
7071   000360
7072   JMP @121
7073   LDA 2 120
7074   LDA 1 123
7075   LDA 0 122
7076   ADD 1 0
7077   STA 0 +0 2

7100   LDA 0 126
7101   LDA 1 125
7102   JSR @+76
7103   000240
7104   NEG 1 1
7105   ADD 1 0
7106   STA 0 126

7107   JSR @+76
7110   000300
7111   JMP 7207
7112   JSR @+50
7113   STA 0 127
7114   STA 1 130

7115   LDA 2 120
7116   LDA 2 +3 2
7117   STA 2 132
7120   JSR @+75
7121   STA 0 131
7122   LDA 1 127
7123   LDA 2 132
7124   JSR @+75
7125   STA 0 132
7126   LDA 1 127
7127   LDA 2 120
7130   LDA 2 +2 2
7131   JSR @+75
7132   LDA 1 131
7133   SUB 1 0
7134   LDA 3 120
7135   LDA 2 +2 3
7136   LDA 1 130
7137   STA 0 +2 3
7140   JSR @+75
7141   LDA 1 132
7142   ADD 1 0
7143   LDA 2 120
7144   STA 0 +3 2
```

```
7145   LDA 1 125
7146   MOVZR 1 1
7147   JSR @+76
7150   000240
7151   NEG 1 1
7152   LDA 0 126
7153   ADD 1 0
7154   JSR @+50
7155   STA 0 127
7156   STA 1 130

7157   LDA 2 120
7160   LDA 2 +5 2
7161   STA 2 132
7162   JSR @+75
7163   STA 0 131
7164   LDA 1 127
7165   LDA 2 132
7166   JSR @+75
7167   STA 0 132
7170   LDA 1 127
7171   LDA 2 120
7172   LDA 2 +4 2
7173   JSR @+75
7174   LDA 1 131
7175   SUB 1 0
7176   LDA 3 120
7177   LDA 2 +4 3
7200   LDA 1 130
7201   STA 0 +4 3
7202   JSR @+75
7203   LDA 1 132
7204   ADD 1 0
7205   LDA 2 120
7206   STA 0 +5 2

7207   LDA 0 +2 2
7210   JSR @+46
7211   004421
7212   004436
7213   LDA 2 120
7214   LDA 0 +4 2
7215   NEG 0 0
7216   JSR @+45
7217   004421
7220   004436
7221   JSR @+45
7222   004000
7223   004061
7224   STA 0 131
```

```
7225   LDA 2 120
7226   LDA 0 +3 2
7227   JSR @+46
7230   004421
7231   004452
7232   LDA 2 120
7233   LDA 0 +5 2
7234   NEG 0 0
7235   JSR @+45
7236   004421
7237   004452
7240   JSR @+45
7241   004000
7242   004141
7243   STA 0 132

7244   LDA 0 126
7245   JSR @+50
7246   STA 0 127
7247   LDA 2 132
7250   JSR @+75
7251   STA 0 132
7252   LDA 1 127
7253   LDA 2 131
7254   JSR @+75
7255   LDA 1 132
7256   ADD 0 1
7257   LDA 2 120
7260   LDA 0 +1 2
7261   STA 1 +1 2

7262   JSR @+76
7263   000300
7264   JMP 7307
7265   JSR @+45
7266   004470
7267   004645
7270   LDA 2 120
7271   LDA 1 123
7272   ADDZL 1 0
7273   STA 0 +2 2
7274   SUB 0 0
7275   STA 0 +3 2
7276   STA 0 +5 2

7277   JSR @+45
7300   004470
7301   004645
7302   LDA 2 120
7303   LDA 1 123
7304   ADDZL 1 0
7305   STA 0 +4 2
7306   JMP @121
```

```
7307    STA 0 130
7310    LDA 0 126
7311    JSR 0+50
7312    LDA 2 130
7313    STA 0 127
7314    JSR @+75
7315    JSR @+45
7316    004000
7317    004221
7320    ADDZL 0 0
7321    JSR @+45
7322    004363
7323    004403
7324    LDA 2 120
7325    STA 0 +3 2
7326    JSR @+47
7327    004373
7330    004403
7331    LDA 2 120
7332    STA 0 +5 2

7333    LDA 1 127
7334    LDA 2 130
7335    JSR @+75
7336    JSR @+45
7337    004000
7340    004301
7341    ADDZL 0 0
7342    JSR @+45
7343    004363
7344    004412
7345    LDA 2 120
7346    STA 0 +2 2
7347    JSR @+47
7350    004373
7351    004412
7352    LDA 2 120
7353    STA 0 +4 2
7354    JMP @121
```

```
6731    107403

6763    001003

7012    JMP @7013
7013    007450

7023    001003

7046    001003

7067    007410

7072    JMP @7067

7360    LDA 1 131
7361    INCR 1 2
7362    ADDZR 0 1
7363    STA 1 131
7364    MOVR 2 2
7365    NEG 0 0
7366    SUBR 0 0
7367    JMP +0 3

7370    LDA 1 132
7371    MOV 1 1 SNR
7372    INC 1 1
7373    INCR 1 0
7374    MOVR 0 0
7375    SUBCR 0 0
7376    ADDZR 0 1
7377    STA 1 132
7400    JMP +0 3

7410    JSR 7370
7411    LDA 1 123
7412    ADD 1 0
7413    JSR @+76
7414    000004
7415    JMP 7424
7416    JSR @+45
7417    004363
7420    004403
7421    JSR @+45
7422    004421
7423    004436
7424    LDA 2 120
7425    STA 0 +1 2
```

```
7426   MOVL 0 0
7427   SUBCR 0 0
7430   JSR 7360
7431   LDA 1 123
7432   JSR @+76
7433   000004
7434   LDA 1 124
7435   ADD 1 0
7436   STA 0 @120

7437   JSR @+47
7440   004373
7441   004403
7442   JSR @+46
7443   004421
7444   004436
7445   JMP @121

7450   STA 0 +1 2
7451   JSR @+76
7452   000003
7453   JMP 7473

7454   MOVL 0 0
7455   SUBCR 0 0
7456   JSR @+76
7457   000001
7460   JSR 7360
7461   LDA 1 124
7462   ADD 1 0
7463   STA 0 @120

7464   LDA 0 122
7465   JSR @+76
7466   000001
7467   JSR 7370
7470   LDA 1 7476
7471   MOVL 0 0 SNC
7472   NEG 1 1

7473   MOV 1 0
7474   JMP @7475
7475   007014
7476   040000
```